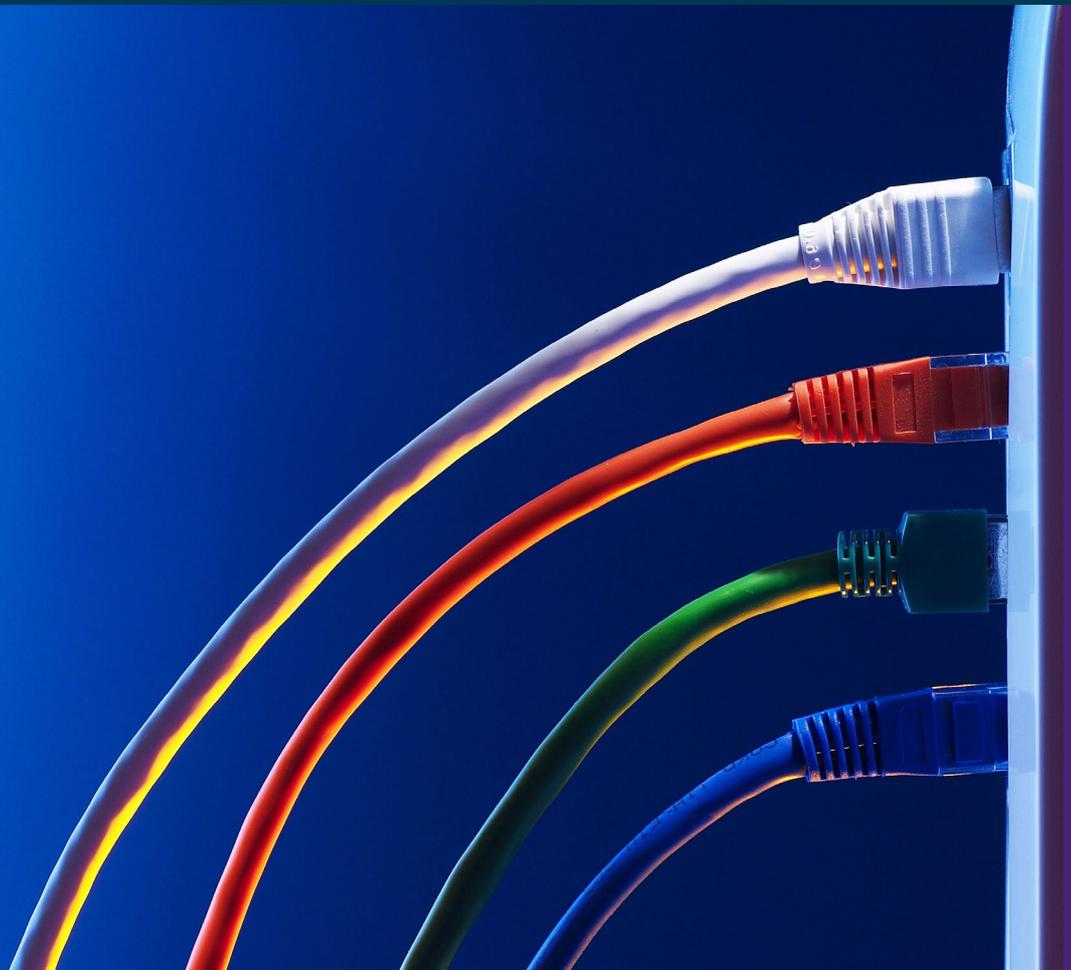


UNIT-e

Handbook - Technical
Document version: 2.2
Issue date: March 2023

All Markets

Managing UNIT-e Web API



For documentation issues email: unit-e.documentation@educationsoftwaresolutions.co.uk

Education Software Solutions Ltd.

Registered office:
11 Kingsley Lodge, 13 New Cavendish Street, London. W1G 9UG.
Registered in England no. 12595779.

www.educationsoftwaresolutions.co.uk

© 2021 Education Software Solutions Ltd. All rights reserved.
No part of this document may be reproduced, photocopied, stored on a retrieval system, translated or transmitted
without the express written consent of ESS Ltd.

Trademark Notice

All company and product names may be trademarks of their respective companies.

4.54 enhancements

The following enhancements are provided in this release:

- *Report method* (see page 51) and *Import endpoint* (see page 59)

If you import a person telephone number with a leading "+", the + is replaced on import with a space. The + sign should be replaced with %2B in the API call to ensure that the + is imported correctly.



Enhancements introduced in previous releases are listed in *Appendix A: Previous enhancements* (see page 130).

Contents



If you are unable to find the information you require, the search facility in Adobe Acrobat enables you to search individual PDFs or the entire UNIT-e documentation catalogue. Refer to the *About UNIT-e documentation* section in the *Working with UNIT-e* handbook ([WWU.pdf](#)) for further details.

	4.54 enhancements	3
Chapter 1:	About UNIT-e Web API	9
Chapter 2:	Setting up UNIT-e Web API	13
	Licence requirements	14
	User access	15
	The web config file user	15
	The Web API user	16
	Installing Web API	19
	Adjusting the web.config file.....	20
	Copying the Capita.lic file	22
	Testing your endpoint calls	23
	Referencing the API	24
	Building user class assemblies.....	25
	Connecting to a test database.....	27
Chapter 3:	Using UNIT-e Web API	29
	Communicating with the API.....	31
	Login method	32
	Class method.....	33
	About System classes	33
	Get endpoint	35
	Delete endpoint	37
	Update endpoint	39
	Update property endpoint.....	41
	Find endpoint	44
	Create endpoint.....	46
	Insert endpoint.....	48
	Lock endpoint.....	50
	Report method	51
	Export endpoint	52
	Export with parameters endpoint	55
	Export with parameters using special characters endpoint	57
	Import endpoint.....	59
	Export Schema endpoint.....	61

Process method	63
EnrolStudent endpoint.....	64
EnrolStudents endpoint	66
Register week pattern updates from register marks endpoint	69
Finance method	71
Pay360InitialisePayment endpoint.....	72
Pay360CompleteEnrolment.....	77
Pay360SetDefaultCard endpoint.....	82
Pay360CompleteRegisterCard endpoint	84
Pay360DeleteCard endpoint.....	86
Pay360InitialiseRegisterCard endpoint.....	88
Security method	90
Create Online Membership endpoint.....	91
Validate Online Membership endpoint.....	93
Update Online Membership endpoint.....	95
Send Password Reset Online Membership endpoint ...	97
Reset Password Online Membership end point	100
InvalidateSession method.....	102
Chapter 4: Testing UNIT-e Web API	103
Prerequisites	104
UNITe API Explorer.....	105
Get class	107
Find class.....	108
Update Property	109
Export Report	110
Export Report with Filter.....	111
Enrol Student	112
Enrol Students.....	113
Pay360 Test pages.....	115
CompleteEnrolment.aspx test page	116
CompleteRegisterCard.aspx test page.....	117
TestApi.aspx test page	118
TestPay360Api.aspx test page	119
Status codes.....	122
Get endpoint	122
Post endpoint	122
Chapter 5: Trouble-shooting and tips	125
Identifying property reference for Insert and Update endpoints.....	126
Importing telephone numbers.....	128
Appendices	129
Appendix A: Previous enhancements.....	130
4.52 enhancements	130
4.51 enhancements	131
4.49 enhancements	131

Glossary	133
Index	135
List of figures	137

Chapter 1:

About UNIT-e Web API

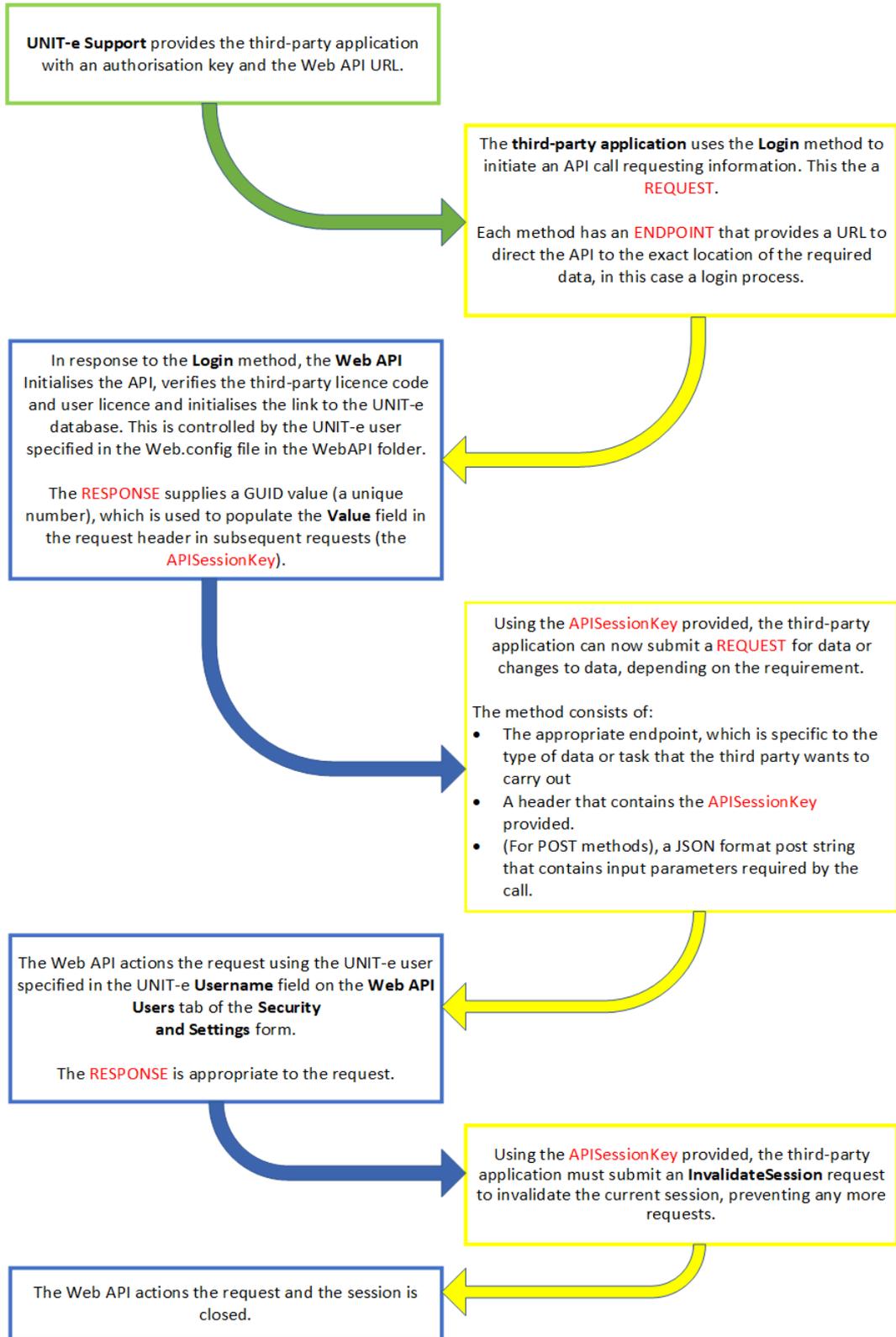
UNIT-e Web API is a tool that provides an interface between third-party applications and a UNIT-e web-based service such as UNIT-e Online Services. By using a Web API you can define a set of functions (e.g. subroutine definitions, protocols, and tools) that enable the third-party software to access data within a UNIT-e application or specific features of a UNIT-e application. For example, a Web API would provide a way for third-party software to access and reference your UNIT-e Student records.

This functionality must be secure. Access to UNIT-e Web API is usually controlled by a Key/Secret combination when calling to the API externally. There is also an option to use LDAP (if licensed) to identify the third party calling the API if the API is installed internally, although this latter situation is less common.

This handbook is designed for MIS Managers with experience of using Web APIs. It explains the setup requirements and gives examples of the methods available to you, including example endpoints and example JSON format POST strings. Sections on testing your endpoints and tips and trouble-shooting are included.

Figure 1 gives an overview of the process.

Figure 1:
The Web API process



This handbook covers the following:

- *Chapter 2: Setting up UNIT-e Web API* (see page 13)
- *Chapter 3: Using UNIT-e Web API* (see page 29)
 - **Login** (see page 32): This initialises the API, verifies the third-party licence code and user licence and initialises the link to the UNIT-e database. This must be called before you make any other calls.
 - **Class** (see page 33): Covers **Get, Delete, Update, Update property, Find, Create, Insert** and **Lock** endpoints for UNIT-e classes.
 - **Report** (see page 51): Covers the use of reports to import data into UNIT-e and export data from UNIT-e.
 - **Process** (see page 63): Covers processes that are normally available within UNIT-e. Endpoints for enrolments and register updates are currently provided.
 - **Finance** (see page 71): Covers processes that deal with Pay360 payments and credit card management.
 - **Security** (see page 90): Covers the creation, validation and update of Online Memberships and the resetting of passwords.
 - **InvalidateSession** (see page 102) ends a session and prevents any more requests using that session.
- *Chapter 4: Testing UNIT-e Web API* (see page 103)
- *Chapter 5: Trouble-shooting and tips* (see page 125)

Chapter 2:

Setting up UNIT-e Web API

The following elements must be set up before you can begin to use UNIT-e Web API:

- *Licence requirements* (see page 14)
- *User access* (see page 15)
- *Installing Web API* (see page 19)
- *Adjusting the web.config file* (see page 20)
- *Copying the Capita.lic file* (see page 22)
- *Testing your endpoint calls* (see page 23)
- *Referencing the API* (see page 24)
- *Building user class assemblies* (see page 25)
- *Connecting to a test database* (see page 27)

Licence requirements

The following are required:

- The Web API (WAPI) licence.
- If you wish to use a third-party product with your UNIT-e system you must have a product-specific licence key. This includes the product code, version and a licence expiry date. This allows third-party product licensing to be controlled in the same way as UNIT-e product licensing.
- Any third-party wishing to develop software to work with the UNIT-e Web API must request an authorisation key for each of their products.

The third-party must provide a version number (in the form '9.9') and product name (in free text) as well as the IP address of the calling system. A product-specific licence key, the version number and the product name are used to generate the authorisation key for that product, which must be embedded into the third-party software to allow it to use the UNIT-e Web API.



Authorisation keys and the URL are available from the Support Team.

User access

There are two different stages to the Web API process, each requiring different UNIT-e user details:

- *The web config file user* (see below)
- *The Web API user* (see page 16)

The web config file user

The first stage of the process uses the UNIT-e user specified in the **web.config** file to make the initial connection to the web service, manage logins and sessions, and send emails to the admin email address if issues are identified.

Create a user in UNIT-e Database Manager (Users) [UDMUS.exe] and assign the user to the **Web API Config User (gn_webapi)** user group. This user group allows the following class access:

- Person message (Select, Insert)
- Person message data (Select, Insert)
- Web API login (all access)
- Web API session (all access)
- Web API session archive (all access)



Refer to *Adjusting the web.config file* (see page 20) for further information.

The Web API user

The second stage of the process deals with completing the request (exporting data, inserting a record, executing a specific finance process etc.). This stage uses the UNIT-e user(s) specified in the **UNIT-e Username** field on the **Web API Users** tab of the **Security and Settings** form.

If you have a number of third parties connecting to the Web API using a number of client ids and secrets, it's good practice to create multiple Web API users in UNIT-e Database Manager (Users) [UDMUS.exe] and list them separately on the **Web API Users** tab. This enables you to closely control access rights.

Each Web API user should have permissions to the following:

- The appropriate method(s) on the **Settings** form (see notes below).
- The UNIT-e classes required by the method(s).



Although it's possible to use the same UNIT-e user in the **web.config** file and on the **Web API Users** tab, this is not recommended. It would mean that the UNIT-e user would potentially need a wide range of permissions to deal with a range of requests, resulting in one third party being able to access classes only required by another third party.



To create a Web API user with API login

1. Open the **Security and Settings** form from the **Tools** menu in UNIT-e Information Interface or from the **Tools** menu in UNIT-e Workplace and go to the **Web API Users** tab (Figure 2).

Figure 2:
Security and
Settings - Web
API Users

Key	Secret	Client Id	Description	UNIT-e Username	IP	Active	LDAP	Settings
WEBAPI	zJUSNzE...		Web API	webapi		<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Create Logon

Key: Description: LDAP

IP Address: UNIT-e Username: SIMS ID

Client Id:

2. Complete the following fields:

Key: This licence authentication key is provided by Support Services. It's unique and is based on the combination of licence key, version and application name. Version and application name are defined by your institution and form part of all request endpoint URLs.



The **Key** MUST be the licence key provided by ESS. Along with the **Secret** value it forms the authorization header. It must also be used in the endpoint URL along with the supplied numeric code.

Description: Add a description.

IP Address: You can add the **IP Address** of the client making the web request, separating multiple addresses with a comma.



The **IP Address** is optional - calls can be made from any machine if no value is entered.



Double-click the **IP Address** field to open it so that you can edit, add or amend values.

UNIT-e Username: Select a UNIT-e user. This user that you specify must have the appropriate permissions to access data. User access requirements are specified against each of the methods in *Chapter 3: Using UNIT-e Web API* (see page 29).

LDAP: This check-box provides the option to login using LDAP (if licensed) to identify the third party calling the API if the API is installed internally. This situation is

uncommon (e.g. a member of staff may wish to connect to the API seamlessly) and you should otherwise leave this check-box unticked.

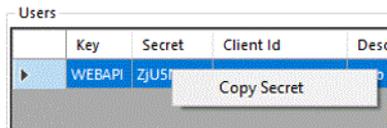
SIMS ID: This check-box isn't currently in use.

3. Click **Add User**. The **Secret** value is generated. You can right-click on this field to copy the value to the clipboard so that it can be applied to the web request header (Figure 3).



You must close and reopen the **Security and Settings** form to make the **Copy Secret** option is available.

Figure 3:
Security and Settings - secret key



4. Click the **Settings** button (Figure 4) to open the **Settings** form (Figure 5).

Figure 4:
Security and Settings - Settings button

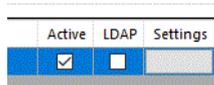
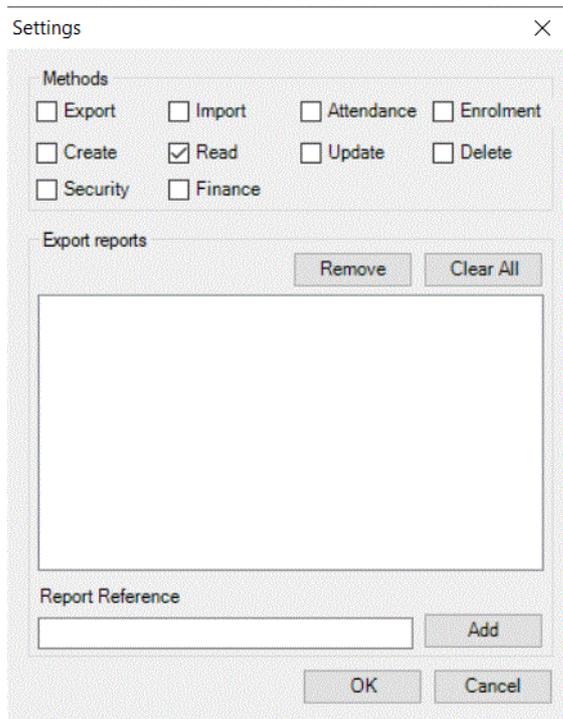


Figure 5:
Security and Settings - Settings



5. The **Settings** form enables you to select the **Methods** available to the user. If the user has access to the **Export** method, use the **Report References** field to add the appropriate export reports.



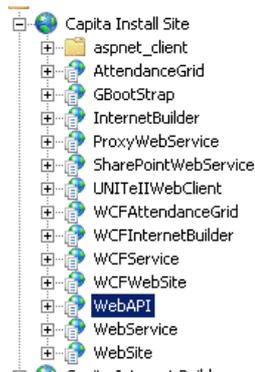
Refer to *Report method* (see page 51) for further information.

Installing Web API

Run the **setup.exe** from the **Software Setup\Web API** folder in the latest image and go through the **Web API Setup Wizard**, noting the following:

- The website must be installed with a certificate using TLS encryption (https), as http unencrypted connections are not supported.
- Web API can be installed in its own website or alongside existing UNIT-e websites, using a .Net 4.x Integrated application pool, as shown in Figure 6.

Figure 6:
Web API
Installation



Adjusting the web.config file

Locate the Web API web.config file in **C:\inetpub\wwwroot\WebAPI** (Figure 7) or select WebAPI in IIS Manager and open the **Application Settings** (Figure 8).

Figure 7:
web.config file

```
<configuration>
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="PreserveLoginUrl" value="true" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
    <add key="Database" value="UNITEDATABASE" />
    <add key="Username" value="UNITEUSERNAME" />
    <add key="Password" value="UNITEPASSWORD" />
    <add key="SessionTimeout" value="2" />
    <add key="DataLimit" value="100" />
    <add key="AdminEmailAddress" value="ADMINISTRATIONEMAILADDRESS" />
    <add key="Debug" value="false" />
    <add key="Scope" value="unit-e:read" />
    <add key="Authority" value="https://simsid-partner-stsserver.azurewebsites.net/" />
    <add key="AuthorisationServerUrl" value = "" />
    <add key="EnableSIMSId" value="false" />
    <add key="EnableAuthService" value="false" />
  </appSettings>
</configuration>
```

Figure 8:
Web API
Application
Settings

 Application Settings

Use this feature to store name and value pairs that managed code applications can use at runtime.

Name	Value	Entry Type
AdminEmailAddress	ADMINISTRATIONEMAILADDRESS	Local
Authority	https://simsid-partner-stsserver.azurewebsites.net/	Local
ClientValidationEnabled	true	Local
Database	UNITEDATABASE	Local
DataLimit	100	Local
Debug	false	Local
EnableSIMSId	false	Local
Password	UNITEPASSWORD	Local
PreserveLoginUrl	true	Local
Scope	unit-e:read	Local
SessionTimeout	2	Local
UnobtrusiveJavaScriptEn...	true	Local
Username	UNITEUSERNAME	Local
webpages:Enabled	false	Local
webpages:Version	3.0.0.0	Local

Amend the following as required:

- **<add key="webpages:Version" value="3.0.0.0" />**
- **<add key="webpages:Enabled" value="false" />**
- **<add key="PreserveLoginUrl" value="true" />**
- **<add key="ClientValidationEnabled" value="true" />**
- **<add key="UnobtrusiveJavaScriptEnabled" value="true" />**
- **<add key="Database" value="{database name}" />**

- `<add key="Username" value="{webapi config user}" />`
- `<add key="Password" value="(webapi config user password)" />`



The 'Username' and 'Password' values must relate to a user with rights to specific UNIT-e classes. Refer to *The web config file user* (see page 15) for further information.

- `<add key="SessionTimeout" value="2" />`

'SessionTimeout' is the length of time (in minutes) that a unique session identifier (GUID - Globally Unique Identifier) stays alive during a period of inactivity. For example, if the setting is '2' and a GUID is supplied via the **Login** endpoint (see page 32), that GUID can be used to call a Web API endpoint and then used again in one and a half minutes time (and again in a one and a half minutes and so on) without timing out. If there's no activity for three minutes before the GUID is used again, the GUID will be invalid and the call will fail.

You can adjust this value to suit your requirements. Once a session has timed out it's moved to the 'Web API session archive' class when the next successful attempt is received.

- `<add key="DataLimit" value="100" />`

This controls the number of records returned. You may adjust this value to suit your requirements but should avoid returning a very large number of records, as this may cause the process to fail.

- `<add key="AdminEmailAddress" value="{admin email address}" />`

This value should be the email address that receives emails when the Web API identifies a potential issue.

For example, when the Web API recognises that a GUID has been used from more than one IP address (an indication that the session may have been hijacked), it invalids that session and sends a warning email to the specified email address, prompting some form of investigation.

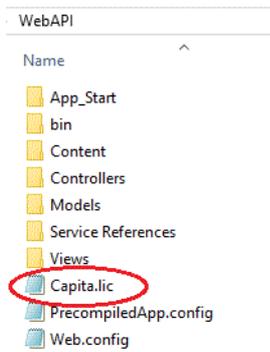
- `<add key="Debug" value="false" />`

This can be set to "true" during testing.

Copying the Capita.lic file

Copy your **Capita.lic** file to the WebAPI website folder as you would do for the standard WCF webservice (Figure 9).

Figure 9:
Capita.lic file in
WebAPI folder



Testing your endpoint calls

We recommend using one of the following to test that endpoint calls are working:

- SoapUI (<https://www.soapui.org>)
- Advanced REST client (a plug-in for Google Chrome)
- Postman (<https://www.postman.com>)



Refer to *Testing UNIT-e Web API* (see page 103) for further information.

Referencing the API

The UNIT-e Web API is referenced via a URL endpoint that's made available either internally or externally on your site.



Authorisation keys and the URL are available from the Support Team.

Building user class assemblies

This process must be carried out if you wish to run Web API processes, IICL processes or both against user classes.



Rights are required to the 'Build user class assemblies' process under the **Database Manager | Class Editor** module and menu item.



To build user class assemblies:

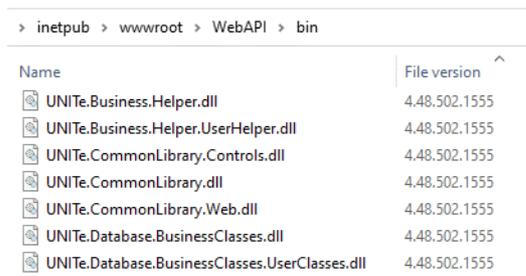
1. In the **Tools** menu in UNIT-e Information Interface, select **Build user classes** to open the **Build user class assemblies** form (Figure 10):

Figure 10:
Build user class assemblies

2. In the **Path for Assemblies** field, enter the path for the (temporary) location in which you wish the assemblies to be built.
3. In the **Strong name key file distributed field** enter the path and filename of the snk file used to sign user class assemblies (e.g. 'C:\Program Files (x86)\Education Software Solutions Ltd\UNIT-e Information Interface\Support Files\unitedemo.snk').
4. Select the **Enable** check box if you wish Web API (and IICL) to load any additional assemblies present for user classes. If not selected, Web API/IICL uses the standard DLLs, as before.
5. Click on the **Build Assemblies** button to generate the DLLs to the location specified.
6. Two logs are provided to allow you to view any errors.
7. Once generated, the user class DLLs must be copied to the **inetpub\wwwroot\WebAPI\bin** folder in Dev and Live.

The version number of the user class DLLs must be the same as the version number (File version) of the UNITE.Database.dll and UNITE.Database.BusinessClasses.dll (Figure 11). For this reason, user class assemblies must be regenerated with every upgrade.

Figure 11:
UNITe.Database.dll and
UNITe.Database.BusinessClasses.dll



Name	File version ^
UNITe.Business.Helper.dll	4.48.502.1555
UNITe.Business.Helper.UserHelper.dll	4.48.502.1555
UNITe.CommonLibrary.Controls.dll	4.48.502.1555
UNITe.CommonLibrary.dll	4.48.502.1555
UNITe.CommonLibrary.Web.dll	4.48.502.1555
UNITe.Database.BusinessClasses.dll	4.48.502.1555
UNITe.Database.BusinessClasses.UserClasses.dll	4.48.502.1555

Connecting to a test database

This section explains how to set up another instance of the Web API to connect to a test database.



To connect to a test database

1. Obtain a **Capita.lic** file with a connection to the UNIT-e test database.
2. Create a separate web site with its own hostname and certificate for https connections.



Refer to *Installing Web API* (see page 19) for details.

3. Copy the whole Web API folder tree structure from the live web site to the test web site.
4. Ensure you have the new **Capita.lic** file at the root of the Test Web API folder.
5. Edit the Test Web API Web.config file and amend the database, username and password values as required.



Refer to *Adjusting the web.config file* (see page 20) for details.

6. Login UNIT-e Information Interface into the UNIT-e Test database and amend the Web API Users settings as required. These should mostly be the same as the settings in the live database.



Refer to *Login method* (see page 32) for details.

7. Connect the test site to the Test Web API URL.

Chapter 3:

Using UNIT-e Web API

UNIT-e Web API provides the following methods:

- *Login method* (see page 32): This method initialises the API, verifies the third-party licence code and user licence and initialises the link to the UNIT-e database. This method must be called before you make any other calls.
- *Class method* (see page 33): This method offers the following endpoints for classes within the UNIT-e database:
 - *Get endpoint* (see page 35)
 - *Delete endpoint* (see page 37)
 - *Update endpoint* (see page 39)
 - *Update property endpoint* (see page 41)
 - *Find endpoint* (see page 44)
 - *Create endpoint* (see page 46)
 - *Insert endpoint* (see page 48)
 - *Lock endpoint* (see page 50)
- *Report method* (see page 51): This method enables you to import and export reports from UNIT-e. The following endpoints are provided:
 - *Export endpoint* (see page 52)
 - *Export with parameters endpoint* (see page 55)
 - *Export with parameters using special characters endpoint* (see page 57)

- *Import endpoint* (see page 59)
- *Export Schema endpoint* (see page 61)
- *Process method* (see page 63): This method is used for processes that are usually available within UNIT-e. The following endpoints are provided:
 - *EnrolStudent endpoint* (see page 64)
 - *EnrolStudents endpoint* (see page 66)
 - *Register week pattern updates from register marks endpoint* (see page 69)
- *Finance method* (see page 71): This method provides finance-related processes. The following endpoints are provided, enabling you to integrate your own online enrolment software with Pay360 to provide payment functionality:
 - *Pay360InitialisePayment endpoint* (see page 72)
 - *Pay360CompleteEnrolment* (see page 77)
 - *Pay360SetDefaultCard endpoint* (see page 82)
 - *Pay360CompleteRegisterCard endpoint* (see page 84)
 - *Pay360DeleteCard endpoint* (see page 86)
 - *Pay360InitialiseRegisterCard endpoint* (see page 88)
 - *Pay360 Test pages* (see page 115)
- *Security method* (see page 90): This method deals with creating, validating and updating Online Memberships and resetting passwords:
 - *Create Online Membership endpoint* (see page 91)
 - *Validate Online Membership endpoint* (see page 93)
 - *Update Online Membership endpoint* (see page 95)
 - *Send Password Reset Online Membership endpoint* (see page 97)
 - *Reset Password Online Membership end point* (see page 100)
- *InvalidateSession method* (see page 102): This method must be called at the end of process to invalidate the current session, preventing any more requests using that session.

All methods within the UNIT-e Web API return an http status code, the requested data or an error message, depending on the success of the call.

Communicating with the API

It is strongly recommended that you follow the procedure below when communicating with the API.



To communicate with the API

1. Call the **Login** endpoint to initialise the API, verify the third-party licence code and user licence and initialise the link to the UNIT-e database.
2. Receive the GUID.
3. Carry out your task.
4. Call the **InvalidateSession** endpoint, to invalidate the current session, preventing any more requests using that session.

Login method

The **Login** method initialises the API, verifies the third-party licence code and user licence and initialises the link to the UNIT-e database. This must be called before you make any other calls.

Login requires the creation of a Web API user on the **Web API Users** tab in the **II Security and Settings** tool.



Refer to *The Web API user* (see page 16) for further details.

Endpoint example

The endpoint is as follows:

{header}/{virtual directory}/login/{licence key}/{version}/{application name}/{licence authentication key}

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/login/API/1.0/Web API Test App 2/123456>

Add the request header 'Authorization' with the value {licence key}:{secret key value}



If you make a Postman call with Authorisation using the Secret only (i.e. the {licence key} is omitted), the result is an Unauthorised response and 'LDAP not enabled' in the Event Viewer (even though you're operating in non-LDAP mode). This may be corrected by entering the {licence key}.

Response

A successful response is the http code 200 and a GUID value, which is used to populate the **Value** in the request header in subsequent requests (Figure 12).

Figure 12:
GUID added to request header

Header	Value
Authorization	49b7ba59-1bf8-438e-99e8-2c47b2c39a70

Class method

The **Class** method covers **Get, Delete, Update, Update property, Find, Create, Insert** and **Lock** endpoints for UNIT-e classes. These can be used with any UNIT-e 'Data' class (i.e. not 'System' classes).

- *About System classes* (see below)
- *Get endpoint* (see page 35)
- *Delete endpoint* (see page 37)
- *Update endpoint* (see page 39)
- *Update property endpoint* (see page 41)
- *Find endpoint* (see page 44)
- *Create endpoint* (see page 46)
- *Insert endpoint* (see page 48)
- *Lock endpoint* (see page 50)



The session identifier must be stored within the 'APISessionKey' header, as the endpoints validates the session identifier against the UNIT-e database before any data is returned.

About System classes

'System' classes that have been specifically excluded from standard business class generation (i.e. those that are NOT normally available via the Class method in the WebAPI) are as follows:

- audit
- auditdetails
- class
- function
- functionparameter
- inputcomponent

- lookup



Although Lookup is excluded it is possible to call class/find/valid to retrieve lookup values.

- reportnode
- reportviewdefinition
- reportxmldefinition
- sequence
- setting
- settingkey
- xmladatadefinition

Most other system classes should be available but you should note that whilst it's possible, updates to most system tables via the WebAPI are not recommended.

Get endpoint

Purpose

The **Get** endpoint enables you to request a record in a UNIT-e class, based on an ID. The class data is returned in JSON format.



Use the **UNITe API Explorer** (see page 105) to test your Get endpoint.

Permissions required

The **Get** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Read** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate class and any parent classes (e.g personaldetails, person).

Endpoint example

The endpoint is as follows:

`https://{host_header}/{virtual directory}/class/get/{class reference}/{class record id}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/class/get/person/999000018338582>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

Response

Example http response in JSON format:

```
{
  "$id": "1",
  "$type": "UNITe.Business.Helper.Person, UNITe.Business.Helper",
  "Reference": "",
  "Surname": "Smith",
  "Forenames": "Jane",
  "Title": "Miss",
  "Gender": "F",
  "DateOfBirth": "1991-01-14T00:00:00",
  "PlaceOfBirth": "",
  "CurrentAddress": "999000018338592",
  "CurrentTelephone": "999000018338593",
  "IsDeceased": false,
  "IsLDAPAccountRequired": true,
  "IsLDAPAccountExists": true,
  "IDCardNumber": "",
  "Stamp": { "$type": "System.Byte[], mscorlib",
    "$value": "AAAAAANXhdE=" },
  "ExtendedProperties": null,
  "DefaultValues": null,
  "Id": "999000018338582"
}
```

Delete endpoint

Purpose

The **Delete** endpoint enables you to delete a record in a UNIT-e class, based on the class name and record ID.



There's no Cascade Delete functionality so you can only delete a record if no child records exist.

Permissions required

The **Delete** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Delete** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Delete** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate UNIT-e class.
- Access to the pick-list report associated with the UNIT-e class.
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for any UNIT-e classes in the pick-list report.
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the 'Lookup valid values' class if any lookups are referenced in the pick-list report.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/class/delete/{class reference}/{class record id}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/class/delete/studentevent/999000037163020>

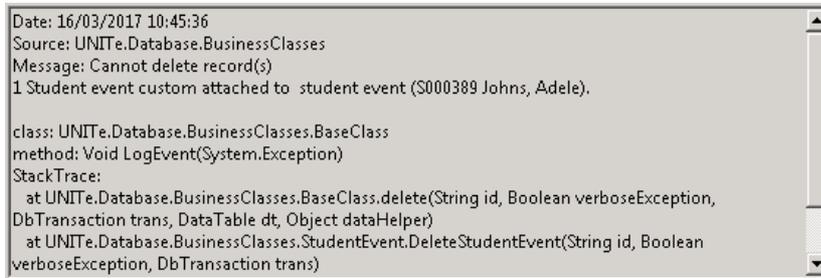
Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

Response

The endpoint returns 'True' if the deletion is successful.

The endpoint returns 'Cannot delete record(s)' if the deletion is unsuccessful, as shown in Figure 13, and logs an entry in the UNIT-e event log on the Web API server.

Figure 13:
API Delete -
response
unsuccessful



```
Date: 16/03/2017 10:45:36
Source: UNITE.Database.BusinessClasses
Message: Cannot delete record(s)
1 Student event custom attached to student event (S000389 Johns, Adele).

class: UNITE.Database.BusinessClasses.BaseClass
method: Void LogEvent(System.Exception)
StackTrace:
   at UNITE.Database.BusinessClasses.BaseClass.delete(String id, Boolean verboseException,
DbTransaction trans, DataTable dt, Object dataHelper)
   at UNITE.Database.BusinessClasses.StudentEvent.DeleteStudentEvent(String id, Boolean
verboseException, DbTransaction trans)
```

Update endpoint

Purpose

The **Update** endpoint enables you to update a record in a in a UNIT-e class, based on the class name and record ID.



Use the **UNITe API Explorer** (see page 105) to test your Update endpoint.

Permissions required

The **Update** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Update** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Update** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate UNIT-e class and any parent classes.

Endpoint example

The endpoint is as follows:

`https://{host_header}/{virtual directory}/class/update/{class reference}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/class/update/student>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

Before carrying out an update you must ALWAYS use the **Get** endpoint to get the record (see page 35). This is because the record is timestamped ("Stamp" in the example below).

The update won't be successful if you attempt to use the **Update** endpoint without first using the **Get** endpoint, even if you have previously updated the same record.

You can use the JSON that's returned as a template, amending the data to create the POST query used by the **Update** endpoint.



When amending the JSON you must retain all the fields in the class, not just those being updated.

```
{
  "$id": "1",
  "$type": "UNITe.Business.Helper.Person, UNITe.Business.Helper",
  "Reference": "",
  "Surname": "Smith",
  "Forenames": "Jane",
  "Title": "Miss",
  "Gender": "F",
  "DateOfBirth": "1991-01-14T00:00:00",
  "PlaceOfBirth": "",
  "CurrentAddress": "999000018338592",
  "CurrentTelephone": "999000018338593",
  "IsDeceased": false,
  "IsLDAPAccountRequired": true,
  "IsLDAPAccountExists": true,
  "IDCardNumber": "",
  "Stamp": {   "$type": "System.Byte[], mscorlib",
    "$value": "AAAAAANe1Gg="  },
  "ExtendedProperties": null,
  "DefaultValues": null,
  "Id": "999000018338582"
}
```

Response

The endpoint returns 'True' if the update is successful. If unsuccessful, the endpoint returns 'Cannot update record(s)' and logs an entry in the UNIT-e event log on the Web API server.

Update property endpoint

Purpose

The **Update property** endpoint enables you to update specific properties within a record, based on the class name, record ID and property reference.

Permissions required

The **Update property** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Update** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Update** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate UNIT-e class and any parent classes.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/class/updateproperty/{class reference}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/class/updateproperty/student>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

```
[
  {
    "Forenames":"Janet",
    "Title":"Miss",
    "Gender":"F",
    "dob":"14-Jan-1991",
    "id":"999000018338582"
  }
]
```

Points to note



Make sure that you refresh the timestamp each time you use the **Update property** endpoint, otherwise the update won't be successful.

- As this is an array, you can enter more than one POST string.
- The POST query uses the property **Reference** displayed in UNIT-e Database Manager (Class Editor) [UDMCE.exe] (Figure 14), unlike the **Update** POST query (see page 39), which uses the whole class record.

Figure 14:
Property refer-
ence

Name	Reference	Type	Link class/Lookup name
Gender	gender	Lookup	Person gender
Date of birth	dob	Date	
Place of birth	pob	Text	
Current address	personaddress	Link	Address (Id)
Current telephone	personteleph...	Link	Telephone (Id)
Deceased	deceased	Flag	
LDAP account required	ldaprequired	Flag	
LDAP account exists	ldapexists	Flag	

- For Boolean values, when setting "true" or "false" you must use a string, for example:

```
= [{"deceased": "true", "id": "1000010151501"}]
= [{"ldaprequired": "true", "id": "1000010151501"}]
= [{"ldapexists": "true", "id": "1000010151501"}]
= [{"safeguardname": "true", "id": "1000010151501"}]
= [{"intcontprefpost": "true", "id": "1000010151501"}]
= [{"intcontprefphone": "true", "id": "1000010151501"}]
= [{"intcontprefemail": "true", "id": "1000010151501"}]
= [{"intcontprefsms": "true", "id": "1000010151501"}]
```

- Unlike other API endpoints, the JSON POST query must be wrapped in square brackets, for example:

```
[{"pob":"Lagos","id":"999000019316498","ldaprequired";"true"}].
```

Response

The endpoint returns 'True' if the update is successful. If unsuccessful, the endpoint returns 'Cannot update record(s)' and logs an entry in the UNIT-e event log on the Web API server.



Oracle only

The JSON Post is case sensitive and should therefore be in a format similar to the following:

```
[{"forenames":"JONATHAN","id":"999000025229039"}]
```

Find endpoint

Purpose

The **Find** endpoint enables you to find records based on the class name and JSON POST query.



Use the **UNITe API Explorer** (see page 105) to test your Find endpoint.

Permissions required

The **Find** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Read** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate class and any parent classes (e.g personaldetails, person).

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/class/find/{class reference}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/class/find/student>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query examples

```
{  
  "forenames":"Jane",
```

```

    "surname":"Smith"
  }
  or
  {
    "forenames":"Jane|John",
    "surname":"Smith|Brown"
  }

```

Response

If the endpoint is successful, there is an http 200 response with output in JSON format, as shown below. All fields in the class are returned, up to the maximum number of records defined in the **web.config** file of the WebAPI website:

<add key="DataLimit" value="100" />



Refer to *Adjusting the web.config file* (see page 20) for further information.

```

[
  {
    "$id": "1",
    "$type": "UNITE.Business.Helper.Student, UNITE.Business.Helper",
    "IsStudent": false,
    "StudentReference": "",
    "PreviousSystemReference": "",
    "DateOfFirstContactAsPotentialStudent": null,
    "DateOfFirstEnrolmentAsActualStudent": null,
    "CurrentHomeAddress": "",
    "CurrentHomeTelephone": "",
    "CurrentLocalAddress": "",
    "CurrentLocalTelephone": "",
  }
]

```

Create endpoint

Purpose

The **Create** endpoint enables you to create a blank instance of a record. You can then use business rules to populate the defined fields. The new record must then be inserted into the database using the **Insert** endpoint (see page 48).

For example, if you use the **Create** endpoint to insert a blank Student record and you have business rules that apply an Id and a Reference, the Id and Reference of the new record will be populated automatically. You can then add further data (date of birth, title etc.) and use the **Insert** endpoint to insert the record into the database.

Permissions required

The **Create** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Create** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Insert** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate class and any parent classes (e.g personaldetails, person).

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/class/create/{class reference}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/class/create/student>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

Response

If the endpoint is successful there is an http 200 response and the details of the record are returned in JSON format, as shown below.

```
[
  {
    "$id": "1",
    "$type": "UNITe.Business.Helper.Student, UNITe.Business.Helper",
    "IsStudent": false,
    "StudentReference": "S058874",
    "PreviousSystemReference": null,
    "DateOfFirstContactAsPotentialStudent": null,
    "DateOfFirstEnrolmentAsActualStudent": null,
    "CurrentHomeAddress": null,
    "CurrentHomeTelephone": null,
    "InvoiceAddress": null,
    "InvoiceTelephone": null,
    "StudentEmailAddress": null,
    "StudentMobileTelephone": null,
    "Tutor": null,
    "TutorGroup": null,
  }
]
```

If unsuccessful the endpoint returns 'Unable to create record' and logs an entry in the UNIT-e event log on the Web API server.

Insert endpoint

Purpose

The **Insert** endpoint enables you to alter and insert the JSON file returned from the **Create** endpoint (see page 46) into the UNIT-e class.

Permissions required

The **Insert** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Create** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Insert** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate class and any parent classes (e.g personaldetails, person).
- **Insert** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the 'Person message data' class in order to process emails that spill over the main 'Person message' Body properties.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/class/insert/{class reference}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/class/insert/student>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

POST query data should be obtained by using the JSON output from the **Create** endpoint (refer to *Create endpoint - Response* (see page 45)).

Response

If successful the endpoint returns 'True'. If unsuccessful the endpoint returns 'Unable to insert record' and logs an entry in the UNIT-e event log on the Web API server.



If you're dealing with a write-back to the 'Person message' class, any email text supplied in the 'Body 1' property that exceeds 2,000 characters is automatically distributed to properties 'Body 2' to 'Body 6' (provided no values have been supplied in these properties) and the 'Person message data' class.

Lock endpoint

Purpose

The **Lock** endpoint enables you to create a lock on a record, based on the class and class record ID.

Permissions required

The **Lock** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Update** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Update** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate UNIT-e class.

Endpoint example

The endpoint is as follows:

`https://{host_header}/{virtual directory}/class/lock/{class reference}/{class record id}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/class/lock/invoice/999000018338583>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

Response

If successful the endpoint returns 'True'. If unsuccessful the endpoint returns 'Unable to lock record' and logs an entry in the UNIT-e event log on the Web API server.

Report method

The **Report** method enables you to export data from UNIT-e and import data into UNIT-e using Report Generator reports. It supports the following endpoints:

- *Export endpoint* (see page 52)
- *Export with parameters endpoint* (see page 55)
- *Export with parameters using special characters endpoint* (see page 57)
- *Import endpoint* (see page 59)
- *Export Schema endpoint* (see page 61)



If you import a person telephone number with a leading "+", the + is replaced on import with a space. The + sign should be replaced with %2B in the API call to ensure that the + is imported correctly.

Export endpoint

Purpose

The **Export** endpoint enables you to extract data from the UNIT-e database by running a UNIT-e Report Generator report using specified run-time parameters. The output is returned in JSON or XML format.

The report must be accessible to the user (i.e. saved in the UNIT-e system and published or owned by the user) and appropriate user rights, filters and masks will also be applied to the report. The user must also have appropriate access rights to the directories specified for the parameter and output files.



Use the **UNITe API Explorer** (see page 105) to test your Export endpoint.

Permissions

The **Export** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Export** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings**, and associated reports (see page 18).
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate UNIT-e class(es).
- Access to the report defined in the URL.
- Access to the directories specified for the parameter and output files.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/report/export/{format}/{report reference}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/report/export/xml/api.test/>



- {format} may be either **xml** or **json**.
- If the report reference contains a period character (.), you must suffix the endpoint with a forward slash '/' (e.g. 'export/xml/api.test/').

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

This POST version of the Export endpoint is provided to show how wildcard searches may be carried out using a % symbol. It allows the parameters to be posted in the body rather than in the query string, as shown in the example URL below.

`https://{host header}/{virtual directory}/report/export/{format}/{report reference}`

<https://localhost/api/report/export/json/student>

Example post data:

student,surname,Ci%

Response

If the endpoint is successful, there is an http 200 response code and data in either JSON format (see below) or XML format (see page 54).

JSON format

```
[
  {
    "c1":999000011646425.0,
    "Student reference number": "SMITH, Jane, Miss (14-Jan-1991)",
  }
  {
    "c1":999000011646430.0,
    "Student reference number": "SMITH, John, Mr (14-Feb-1991)",
  }
]
```

XML format

```
<NewDataSet>
  <student>
    <c1>999000011646425</c1>
    <Student_x0020_reference_x0020_number> SMITH, Jane, Miss (14-Jan-1991) </
    Student_x0020_reference_x0020_number>
  </student>
  <student>
    <c1>999000011646430</c1>
    <Student_x0020_reference_x0020_number> SMITH, John, Mr (14-Feb-1991) </
    Student_x0020_reference_x0020_number>
  </student>
```

Export with parameters endpoint

Purpose

The **Export with parameters** endpoint enables you to extract specific class properties and values from the UNIT-e database by running a UNIT-e Report Generator report using specified run-time parameters. Multiple sets of parameters can be defined, provided you specify a class, property and value, comma-separated.



Use the **UNITe API Explorer** (see page 105) to test your Export with parameters endpoint.

Permissions

The **Export with parameters** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Export** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings**, and associated reports (see page 18).
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate UNIT-e class(es).
- Access to the report defined in the URL.

Endpoint example

The endpoint is as follows:

```
https://{host header}/{virtual directory}/report/export/{format xml/json}/{report reference}/{class},{property},{value}
```

For example:

```
https://webapiinstallsite.unitetest.zone/WebAPI/report/export/xml/student/student,surname,Smith
```



The {value} can be entered using wildcards (e.g. 'Be%').

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

Response

If the endpoint is successful there is an http 200 response code and data in either JSON format or XML format.

JSON format

```
[
  {
    "c1":999000011646425.0,
    "Student reference number": "SMITH, Jane, Miss (14-Jan-1991)",
  }
  {
    "c1":999000011646430.0,
    "Student reference number": "SMITH, John, Mr (14-Feb-1991)",
  }
]
```

XML format

```
<NewDataSet>
  <student>
    <c1>999000011646425</c1>
    <StudentReferenceNumber> S000569 SMITH, Jane, (14-Jan-1991) </
    StudentReferenceNumber>
  </student>
  <student>
    <c1>999000011646430</c1>
    <StudentReferenceNumber> S000570 SMITH, John, (14-Feb-1991) </
    StudentReferenceNumber>
  </student>
</NewDataSet>
```

Export with parameters using special characters endpoint

Purpose

The **Export with parameters using special characters** endpoint enables you to extract specific class properties and values from the UNIT-e database - including special characters - by running a UNIT-e Report Generator report using specified run-time parameters. Multiple sets of parameters can be defined, provided you specify a class, property and value, comma-separated.

Permissions

The **Export with parameters using special characters** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Export** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate UNIT-e class(es).
- Access to the report defined in the URL.

Endpoint example

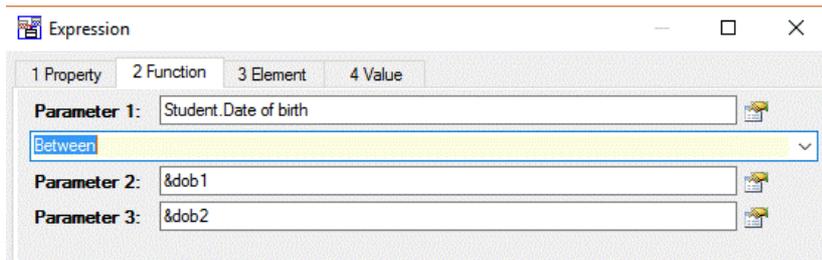
The endpoint is as follows:

`https://{host header}/{virtual directory}/report/export/{format xml|json}/{report reference}/special,{special variable},{value},special,{special variable},{value}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/report/export/xml/studentspecial/special,dob1,01-jan-2000,special,dob2,31-dec-2000>

Figure 15:
Export with
parameters
using special
characters -
report details



The {value} can be entered using wildcards (e.g. 'Be%') and multiple variables can be defined if separated by a comma (e.g. special,{special variable},{value}, special,{special variable},{value}).

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

Response

If the endpoint is successful there is an http 200 response code and data in either JSON format or in XML format.

XML format

```
<NewDataSet>
  <studentspecial>
    <Surname>SMITH</Surname>
    <Forenames>Jane</Forenames>
    <DateOfBirth>14-Jan-1991</DateOfBirth>
  </studentspecial>
</NewDataSet>
```

JSON format

```
[
  {
    "Surname": "SMITH",
    "Forenames": "Jane",
    "DateOfBirth": "14 Jan 1991",
  }
]
```

Import endpoint

Purpose

The **Import** endpoint enables you to import data into the UNIT-e database using a UNIT-e Report Generator report and a POST endpoint. The posted data can be in JSON or XML format.



If you import a person telephone number with a leading "+", the + is replaced on import with a space. The + sign should be replaced with %2B in the API call to ensure that the + is imported correctly.

Permissions

The **Import** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Import** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate UNIT-e class(es).
- Access to the report defined in the URL.

Endpoint example

The endpoint is as follows:

```
https://{host_header}/{virtual directory}/report/import/{post query format}/{UNITe report reference}/
```

For example:

```
https://webapiinstallsite.unitetest.zone/WebAPI/report/import/json/api.import/
```



If the report reference in the endpoint contains a period (.) it must be suffixed with a forward slash '/'.

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

JSON Post query example for adding new records

```
{ "UNITE": { "api.import": [ { "student.forenames": "Jane", "student.surname": "Smith",  
"student.dob": "14-Jan-1991", "student.student": "-1" } , { "student.forenames": "John",  
"student.surname": "Smith", "student.dob": "14-Feb-1991", "student.student": "-1" } ] } }
```

JSON Post query example for updating existing records

```
{ "UNITE": { "api.import": [ { "student.id": "999000037232009", "student.dob": "17-Aug-  
2000" }, { "student.id": "999000037232007", "student.dob": "17-Aug-1999" } ] } }
```

XML Post query example for adding new records (no line breaks in string)

```
<UNITE><api.import><student.forenames>Jane</  
student.forenames><student.surname>Smith</student.surname><student.dob>14-Jan-  
1991</student.dob><student.student>-1</student.student></  
api.import><api.import><student.forenames>John</  
student.forenames><student.surname>Smith</student.surname><student.dob>14-Jan-  
1991</student.dob><student.student>-1</student.student></api.import></UNITE>
```

XML Post query example for updating existing records (no line breaks in string)

```
<UNITE><api.import><student.id>999000037232083</student.id><student.dob>17-  
Aug-2000</student.dob></api.import><api.import><student.id>999000037232082</  
student.id><student.dob>17-Aug-1999</student.dob></api.import></UNITE>
```

When a successful import takes place you receive a status of 0:

```
<UNITE><status>0</status></UNITE>
```

Export Schema endpoint

Purpose

The **Export Schema** endpoint gets an XML schema that has been created within UNIT-e XML Exporter, based on the schema name, report name and query parameters. This call only returns in XML format.

Permissions

The **Export Schema** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Export** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the appropriate UNIT-e class(es).
- Access to the report defined in the URL.

Endpoint example

The endpoint is as follows:

```
https:// {host header}/{virtual directory}/report/exportschema/{schema name}/{report reference}/{class reference},{property name},{value}
```

For example:

```
https://webapiinstallsite.unitetest.zone/WebAPI/report/exportschema/api.unitetest/lp.template.student/student,surname,smith
```

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

Response

An example response is shown in Figure 16.

Figure 16:
API Export
Schema
response

```
HTTP/1.1 200 OK
Cache-Control: no-cache,no-cache
Pragma: no-cache
Content-Length: 369481
Content-Type: application/xml; charset=utf-8
Expires: -1
Server: Microsoft-IIS/7.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
X-XSS-protection: 1
X-Content-Type-Options: nosniff
Content-Security-Policy: script-src 'self'
X-Frame-Options: deny
Strict-Transport-Security: max-age=16070400; includeSubDomains
Date: Wed, 12 Apr 2017 11:52:08 GMT

<?xml version="1.0" encoding="utf-8"?><WebApiUnitTest><lp.template.student><ID>1000010153560</ID><FORENAMES> John.</FORENAMES><SURNAME>
```

Process method

The **Process** method is used for processes that are generally available within UNIT-e (e.g. student enrolment). This method currently supports the following:

- *EnrolStudent endpoint* (see page 64)
- *EnrolStudents endpoint* (see page 66)
- *Register week pattern updates from register marks endpoint* (see page 69)

EnrolStudent endpoint

Purpose

The **EnrolStudent** endpoint enrolls a student using the Student Id, Module Id of the Module on which the student wishes to be enrolled and the Fee band reference.



Use the **UNITe API Explorer** (see page 105) to test your EnrolStudent endpoint.

Permissions

The **EnrolStudent** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Enrolment** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for 'Module' (including extended classes), 'Module fee', 'Offering', 'Student' (including extended classes) and 'Unit' classes.
- **Insert** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for 'Enrolment' (including extended classes), 'Enrolment fee' and 'Fee' classes.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/process/enrolstudent/{student id}/{module id}/
{feeband reference}`

For example:

[https://webapiinstallsite.unitetest.zone/WebAPI/process/enrolstudent/999000037226987/
999000030872952/stdn](https://webapiinstallsite.unitetest.zone/WebAPI/process/enrolstudent/999000037226987/999000030872952/stdn)

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

Response

The EnrolStudent response is a Boolean (True/False).

EnrolStudents endpoint

Purpose

The **EnrolStudents** endpoint creates multiple enrolments based on the student id, module id and feeband, and optionally updates the status and online reference properties if provided in a single call. As this is an array, multiple parameters can be entered within a single call.



- Only one Status, Online Reference and Fee band may be specified.
- If a module has a mandatory child module, the student will automatically be enrolled on that child module.



Use the **UNITe API Explorer** (see page 105) to test your EnrolStudents endpoint.

Permissions

The **EnrolStudent** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Enrolment** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for 'Module' (including extended classes), 'Module fee', 'Offering', 'Student' (including extended classes) and 'Unit' classes
- **Insert** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for 'Enrolment' (including extended classes), 'Enrolment fee' and 'Fee' classes

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/process/enrolstudents`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/process/enrolstudents>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example - single student, single module

This is a POST query example for adding a single student with a single module.

```
[{ "StudentID":["999000037226987"],"ModuleID":["999000030872952"],
  "Status":"1","Onlinereference":"OLREF4","FeeBand":"stnd" }]
```

Response

```
[{"StudentId" : "999000037226987","Enrolment" : [{"ModuleId" : "999000030872952",
  "EnrolmentId" : "999000037232132"}]}
```

POST query example - multiple students, single module

This is a POST query example for adding multiple students with a single module (no line breaks).

```
[{ "StudentID":["999000037226988",
  "999000037227069"],"ModuleID":["999000030872952"],"Status":"1","Onlinereference":"
  OLREF4","FeeBand":"stnd" }]
```

Response

```
[{"StudentId" : "999000037226988","Enrolment" : [{"ModuleId" : "999000030872952",
  "EnrolmentId" : "999000037232136"}]},{"StudentId" : "999000037227069","Enrolment" :
  [{"ModuleId" : "999000030872952", "EnrolmentId" : "999000037232139"}]}
```

POST query example - single student, multiple modules

This is a POST query example for adding a single student with multiple modules.

```
[{
  "StudentID":["999000037226987"],"ModuleID":["999000037227099","999000030872952
  "],"Status":"1","Onlinereference":"OLREF4","FeeBand":"stnd" }]
```

Response

```
[{"StudentId" : "999000037226987", "Enrolment" : [{"ModuleId" : "999000037227099", "EnrolmentId" : "999000037232142"}, {"ModuleId" : "999000030872952", "EnrolmentId" : "999000037232144"}]}
```

Post query example multiple students, multiple modules

This is a POST query example for adding multiple students with multiple modules.

```
[{ "StudentID":["999000037226987", "999000037226988", "999000037227069"], "ModuleID":["999000037227099", "999000030872952"], "Status": "1", "Onlinereference": "OLREF4", "FeeBand": "stnd" }]
```

Response

```
[{"StudentId" : "999000037226987", "Enrolment" : [{"ModuleId" : "999000037227099", "EnrolmentId" : "999000037232147"}, {"ModuleId" : "999000030872952", "EnrolmentId" : "999000037232149"}]}, {"StudentId" : "999000037226988", "Enrolment" : [{"ModuleId" : "999000037227099", "EnrolmentId" : "999000037232152"}, {"ModuleId" : "999000030872952", "EnrolmentId" : "999000037232154"}]}, {"StudentId" : "999000037227069", "Enrolment" : [{"ModuleId" : "999000037227099", "EnrolmentId" : "999000037232157"}]}
```

Register week pattern updates from register marks endpoint

Purpose

The **Register week pattern updates from register marks** endpoint accepts a register id (student, staff, room), a date, and a register mark value. It updates the Register mark record based on the id/date/mark values provided and updates the Student, Staff or Room register week pattern with the new mark in the correct string location. It also validates the Activity week pattern on the date provided to avoid storing invalid marks in the student/staff/room week pattern.

Permissions

The **Register week pattern updates from register marks** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Security** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Update** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for 'Register marks', 'Register marks room', 'Register marks staff' and 'Week pattern'.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/process/updateregistermark`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/process/updateregistermark>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

```
{ "id": "1660000002222508", "type": "student", "date": "11-Sep-2021", "mark": "O" }
```



- The "type" can be "student", "staff" or "room".
- The "id" and "type" are related, as follows:
 - If "type" is "student", the "id" property must be the id of the student register record to be updated.
 - If "type" is "staff", the "id" property must be the id of the staff register record to be updated.
 - If "type" is "room", the "id" property must be the id of the room register record to be updated.

Response

OK (HTTP status code 200) if successful or Bad request (HTTP status code 400) if the supplied data is incorrect.

Finance method

These endpoints enable you to integrate your own online enrolment software with Pay360, providing payment functionality. This means that when a Student enrolls on a Module with associated fees you can pass the information to Pay360, resulting in the creation of an invoice, invoice lines, receipt and receipt lines in your UNIT-e database.



Test pages are provided to enable you to test the processes. Refer to *Pay360 Test pages* (see page 115) for further information.

This section includes the following:

- *Pay360InitialisePayment endpoint* (see page 72)
- *Pay360CompleteEnrolment* (see page 77)
- *Pay360SetDefaultCard endpoint* (see page 82)
- *Pay360CompleteRegisterCard endpoint* (see page 84)
- *Pay360DeleteCard endpoint* (see page 86)
- *Pay360InitialiseRegisterCard endpoint* (see page 88)

Pay360InitialisePayment endpoint

Purpose

The **Pay360InitialisePayment** endpoint initiates the process for payments, generates the redirect URL, a unique reference and an SCP reference.

Permissions

The **Pay360InitialisePayment** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Enrolment** and **Finance** methods on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- The user must be a member of the Online Services web config user (os_online) user group.

Endpoint example

The endpoint is as follows:

https://{host header}/{virtual directory}/finance/Pay360InitialisePayment

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/finance/Pay360InitialisePayment>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example - single enrolment

```
{
  "BackUrl": "",
  "Email": "",
  "FundCode": "",
  "FundName": "",
  "EnrolmentModule": [
    {
      "ModuleID": ""
    }
  ],
  "EnrolmentFees": [
    {
      "FeeID": ""
    }
  ],
  "ReturnUrl": "",
  "StudentForenames": "",
  "StudentReference": "",
  "StudentSurname": ""
}
```

BackUrl: The page to return to if you cancel.

Email: Student email address (student.studentemailaddress)

FundCode: This is mandatory. Used to identify the payment source to enable payment grouping. Defaults to '1' (Online Services).

FundName: The payment description. This appears on the receipt and is definable by your institution.

ModuleID: module.id

FeeID: modulefee.id

ReturnUrl: The page that Pay360 returns to.

StudentForenames: Student.forenames

StudentReference: Student.studentreference

StudentSurname: Student.surname

For example:

```
{"BackUrl": "https://uniteinstitution/WCFWebsite/OnlineServices/complete.aspx",  
  "Email": "test.student123@esstest.co.uk",  
  "FundCode": "1",  
  "FundName": "Student Enrolment",  
  "EnrolmentModule": [{  
    "ModuleID": "999000077980678"  
  }],  
  "EnrolmentFees": [{  
    "FeeID": "999000077980679"  
  }],  
  "ReturnUrl": "https://uniteinstitution/WCFWebSite/OnlineServices/  
  CompleteEnrolment.aspx",  
  "StudentForenames": "test",  
  "StudentReference": "S059121",  
  "StudentSurname": "student"}
```

POST query example - multiple enrolments

```
{
  "BackUrl": "",
  "Email": "",
  "FundCode": "",
  "FundName": "",
  "EnrolmentModule": [
    {
      "ModuleID": ""
    },
    {
      "ModuleID": ""
    }
  ],
  "EnrolmentFees": [
    {
      "FeeID": ""
    },
    {
      "FeeID": ""
    }
  ],
  "ReturnUrl": "",
  "StudentForenames": "",
  "StudentReference": "",
  "StudentSurname": ""
}
```

Response

The Pay360InitialisePayment endpoint returns the following JSON.



The values shown in the following example are for reference only.

```
{
  "RedirectUrl": "https://sbsctest.e-paycapita.com:443/scp/scpcli?ssk=pr239rrinhl5gxwuqcr0vqyr41g6bnd",
  "UniqueReference": "EUZ7227784",
  "ScpReference": "6s84glivr12vijp9qmcjklhnlewccv",
  "ItemDetails": [{
    "reference": "S059121:TMP3D:Tuition Fees",
    "reference2": "Tuition Fees:Test Module Pay 360 API",
    "fundcode": "1",
    "amount": "14400",
    "lineId": "999000077980679"
  }],
  "Status": "SUCCESS"
}
```

RedirectUrl: Pay360 URL.

UniqueReference: Generated GUID.

ScpReference: Generated.

reference: Concatenation of Student ref and fee type.

fundcode: Used to identify the payment source to enable payment grouping. Defaults to '1' (Online Services)

amount: The value of the payment.

lineId: modulefee.id

When you click the **Call API** button you're redirected to the CompleteEnrolment.aspx.



Refer to *Pay360CompleteEnrolment* (see page 77).

Pay360CompleteEnrolment

Purpose

The **Pay360CompleteEnrolment** endpoint verifies payments and creates Enrolments, Invoices, Invoice lines, Receipts and Receipt lines.

The POST query that you use depends on whether you're dealing with a single enrolment and single fee, single enrolment with multiple fees or multiple enrolments with multiple fees. Examples for each are given.



Use the **Pay360 Test pages** (see page 115) to test your EnrolStudent endpoint.

Permissions

The **Pay360CompleteEnrolment** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Enrolment** and **Finance** methods on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- The user must be a member of the Online Services web config user (os_online) user group.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/finance/Pay360CompleteEnrolment`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/finance/Pay360CompleteEnrolment>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example - single enrolment, single fee

Requires *POST query example - single enrolment* (see page 73) plus:

```
{ "Market": "General",  
  "EnrolmentFeeDetails": [{  
    "EnrolmentFees": [{  
      "FeeID": "999000077980679"  
    }]  
  }],  
  "Source": "Source",  
  "FeeBand": "online",  
  "StudentID": "999000077980691"  
}
```

Where the following are required:

Market: Refers to extended classes:

- ISR = 0
- WSR = 1
- FES = 2
- HESA = 3
- General = 4
- HEA = 5

FeeID: modulefee.id

Source: moduleenrolment.source

FeeBand: By default must match the fee band specified against the 'Application\OSEN\Payment\FeeBand' setting key.

POST query example - single enrolment, two fees

Requires *POST query example - single enrolment* (see page 73) plus:

```
{ "Market": "General",
  "EnrolmentFeeDetails": [{
    "EnrolmentFees": [{
      "FeeID": "999000076530966"
    },
    {
      "FeeID": "999000077936676"
    }
  ]
}],
  "Source": "Source",
  "FeeBand": "online",
  "StudentID": "999000077980691"
}
```

POST query example - two enrolments, single fees

Requires *POST query example - multiple enrolments* (see page 75) plus:

```
{ "Market": "General",
  "EnrolmentFeeDetails": [{
    "EnrolmentFees": [{
      "FeeID": "999000077936711"
    },
    {
      "FeeID": "999000077936712"
    }
  ]
}],
  "Source": "Source",
  "FeeBand": "online",
  "StudentID": "999000077980691"
}
```

POST query example - including a sundry item

Requires *POST query example - single enrolment* (see page 73) or *POST query example - multiple enrolments* (see page 75) plus:

```
"SundryItems": [{
  "ItemId": "236",
  "TotalPrice": "898",
  "Name": "SundryItem1",
  "Reference": "8528",
  "Quantity": "1"
}],
```

Response

Enrolments, receipts and invoices are created and the message "Enrolment created successfully" is displayed.

Checking data

Check that Enrolment, invoice and receipt details appear in UNIT-e Finance Manager. You can also verify the details using the following queries in UNIT-e Report Generator if you have knowledge of SQL server management studio.

Enrolments

```
select * from capd_moduleenrolment where e_module = ModuleId
```

Invoices

```
select * from capd_invoice where i_id in (select il_parent from capd_invoiceline where il_id in (select eil_id from capd_enrolmentinvoiceline where eil_invlineenrolment = EnrolmentId))
```

Receipts

```
select * from capd_receipt where r_id in (select rl_parent from capd_receiptline where rl_receiptlineinvoice in (select i_id from capd_invoice where i_id in (select il_parent from capd_invoiceline where il_id in (select eil_id from capd_enrolmentinvoiceline where eil_invlineenrolment = EnrolmentId)) ))
```

Pay360SetDefaultCard endpoint

Purpose

The **Pay360SetDefaultCard** endpoint marks the default card when multiple cards have been added by the student.

Permissions

The **Pay360SetDefaultCard** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Enrolment** and **Finance** methods on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- The user must be a member of the Online Services web config user (os_online) user group.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/finance/Pay360SetDefaultCard`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/finance/Pay360SetDefaultCard>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

The following is required by Pay360.

```
{
  "CardId" : "999999999999"
}
```

Response

```
{  
  'Status' : 'SUCCESS',  
}
```

Pay360CompleteRegisterCard endpoint

Purpose

The **Pay360CompleteRegisterCard** endpoint checks whether the card details have been entered successfully with reference to the ScpReference passed to it. A new record is created in UNIT-e if the details are entered successfully.



Use the **Pay360 Test pages** (see page 115) to test your Pay360CompleteRegisterCard endpoint.

Permissions

The **Pay360CompleteRegisterCard** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Enrolment** and **Finance** methods on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- The user must be a member of the Online Services web config user (os_online) user group.

Endpoint example

The endpoint is as follows:

https://{host header}/{virtual directory}/finance/Pay360CompleteRegisterCard

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/finance/Pay360CompleteRegisterCard>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

The following is required by Pay360.

```
{  
  'ScpReference':'br4inlweew136uy6l7rfi7lapzso1j5',  
}
```

Response

```
{  
  "Status":"SUCCESS",  
  "CardLastFourDigits":"6721",  
  "ExpiryDate":"1220"  
}
```

Pay360DeleteCard endpoint

Purpose

The **Pay360DeleteCard** endpoint marks the credit card as removed, with reference to the CardId passed to it.

Permissions

The **Pay360DeleteCard** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Enrolment** and **Finance** methods on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- The user must be a member of the Online Services web config user (os_online) user group.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/finance/Pay360DeleteCard`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/finance/Pay360DeleteCard>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

The following is required by Pay360.

```
{  
  'CardId':'999999999999'  
}
```

Response

```
{  
  Status: 'SUCCESS'  
}
```

Pay360InitialiseRegisterCard endpoint

Purpose

The **Pay360InitialiseRegisterCard** endpoint exposes the Online Services Credit Card Instalment method to the WebAPI. It allows a student to select **Pay later** and enter their card details, which are stored by Pay360 with a cross referencing token held in UNIT-e, enabling the Stored Card Batch Processing service to run in the back office.

It creates a reference record at Pay360 for the student details passed to it and responds with a RedirectUrl where the student can save their card details. Once the student has entered their card details successfully, SCP redirects back to the returnUrl passed with the request.

Permissions

The **Pay360InitialiseRegisterCard** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Enrolment** and **Finance** methods on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- The user must be a member of the Online Services web config user (os_online) user group.

Endpoint example

The endpoint is as follows:

https://{host header}/{virtual directory}/finance/Pay360InitialiseRegisterCard

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/finance/Pay360InitialiseRegisterCard>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

The following is required by Pay360.

```
{
  'StudentID' : '99999999999',
  'StudentEmail' : 'xyz.test.com',
  'ReturnUrl' : 'https://uniteinstitution/WCFWebsite/OnlineServices/complete.aspx'
}
```

Response

```
{
  'Status' : 'SUCCESS',
  'ScpReference' : 'br4inlweew136uy6l7rfi7lapzso1j5',
  'RedirectUrl' : 'https://sbsctest.e-paycapita.com:443/scp/scpli?ssk=srgj23k63zxhy04zifaf4218ps4no0g'
}
```

Security method

These endpoints deal with the creation, validation and updating of Online Memberships and the resetting of passwords.

- *Create Online Membership endpoint* (see page 91)
- *Validate Online Membership endpoint* (see page 93)
- *Update Online Membership endpoint* (see page 95)



If you use any of these security calls (create/change/validate online membership) and no access to the calls is provided in II, you receive a 401 error and an Event Log message advising you of missing permissions (e.g. 'Cannot create online membership account - 'Security' method not enabled').

- *Send Password Reset Online Membership endpoint* (see page 97)
- *Reset Password Online Membership end point* (see page 100)

Create Online Membership endpoint

Purpose

The **Create Online Membership** endpoint creates an Online membership record for an existing student with a linked Online membership audit record. It also increments the id counter ready for the next record to be created.

Permissions

The **Create Online Membership** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Security** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Insert** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the 'Online membership' class.
- **Insert** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the 'Online membership audit' class.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/security/onlinemembership/create`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/security/onlinemembership/create>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

Post query example for creating online membership record:

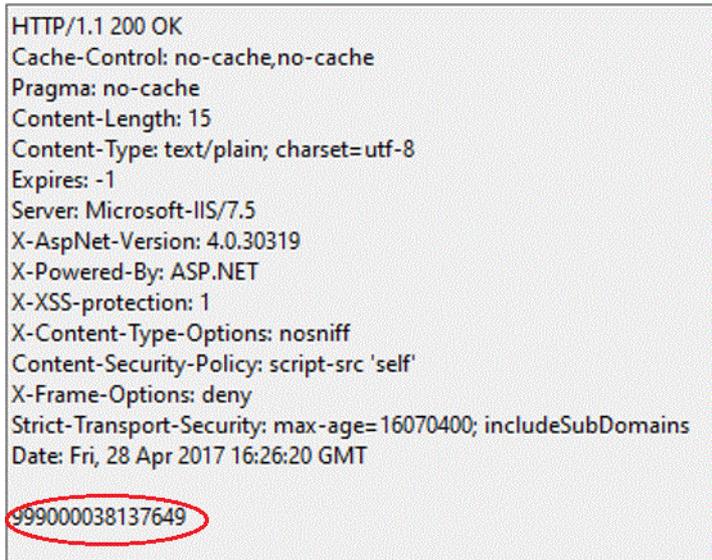
```
{  
  "studentid": "123456789",  
  "email": "test@test.com",  
  "password": "password",  
  "ipaddress": "127.0.0.1"  
}
```

Response

If unsuccessful this endpoint returns an http 401 unauthorised response and detailed event log error message.

If successful this endpoint returns the id of the new online membership record, as shown in Figure 17.

Figure 17:
API Create
Online Mem-
bership
response



```
HTTP/1.1 200 OK  
Cache-Control: no-cache,no-cache  
Pragma: no-cache  
Content-Length: 15  
Content-Type: text/plain; charset=utf-8  
Expires: -1  
Server: Microsoft-IIS/7.5  
X-AspNet-Version: 4.0.30319  
X-Powered-By: ASP.NET  
X-XSS-protection: 1  
X-Content-Type-Options: nosniff  
Content-Security-Policy: script-src 'self'  
X-Frame-Options: deny  
Strict-Transport-Security: max-age=16070400; includeSubDomains  
Date: Fri, 28 Apr 2017 16:26:20 GMT  
999000038137649
```

Validate Online Membership endpoint

Purpose

The **Validate Online Membership** endpoint validates an Online membership record for an existing student and creates a linked Online membership audit record. The id of the Online membership is returned if the search is successful.

Permissions

The **Validate Online Membership** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Security** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Read** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the 'Online membership' class.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/security/onlinemembership/validate`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/security/onlinemembership/validate>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

Post query example for validating online membership record:

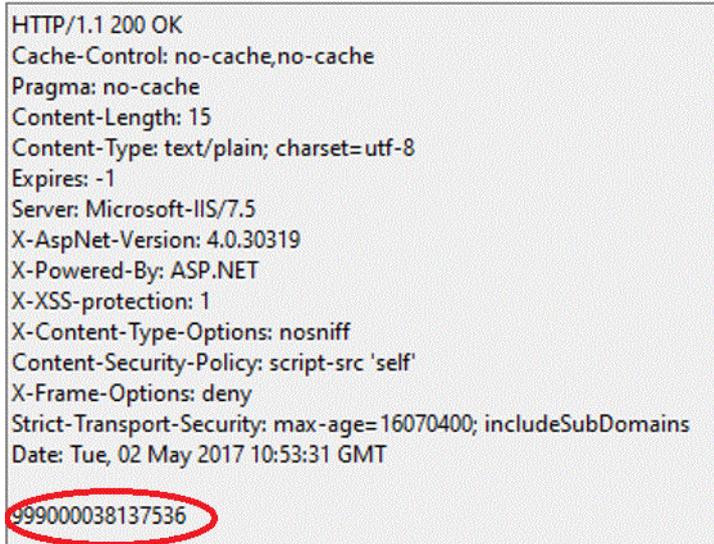
```
{  
  "email": "test@test.com",  
  "password": "password",  
  "ipaddress":  
    "127.0.0.1"  
}
```

Response

If unsuccessful this endpoint returns an http 401 unauthorised response and detailed error in the event log.

If successful this endpoint returns the id of the online membership record, as shown in Figure 18.

Figure 18:
API Validate
Online Mem-
bership
response



```
HTTP/1.1 200 OK  
Cache-Control: no-cache,no-cache  
Pragma: no-cache  
Content-Length: 15  
Content-Type: text/plain; charset=utf-8  
Expires: -1  
Server: Microsoft-IIS/7.5  
X-AspNet-Version: 4.0.30319  
X-Powered-By: ASP.NET  
X-XSS-protection: 1  
X-Content-Type-Options: nosniff  
Content-Security-Policy: script-src 'self'  
X-Frame-Options: deny  
Strict-Transport-Security: max-age=16070400; includeSubDomains  
Date: Tue, 02 May 2017 10:53:31 GMT  
999000038137536
```

Update Online Membership endpoint

Purpose

The **Update Online Membership** endpoint updates an Online membership record for an existing student. 'True' is returned if the update is successful.

Permissions

The **Update Online Membership** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Security** method on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Update** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the 'Online membership' class.

Endpoint example

The endpoint is as follows:

https://{host header}/{virtual directory}/security/onlinemembership/changepassword

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/security/onlinemembership/changepassword>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example

Post query example for updating online membership record:

```
{  
  "email": "test@test.com",  
  "oldpassword": "password",  
  "newpassword": "password1"  
}
```

Response

If unsuccessful the endpoint returns an http 401 unauthorised response and detailed event log error message.

If successful the endpoint returns `True`.

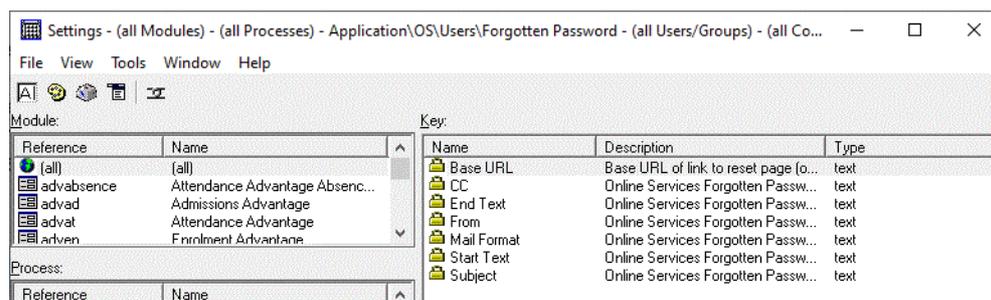
Send Password Reset Online Membership endpoint

Purpose

The **Send Password Reset Online Membership** endpoint generates an email for a student containing a URL that enables them to reset their password. A record is added to the 'Person message' class, which is then sent to the student by the UNIT-e Windows Services service.

The setting 'Application\OS\Users\Forgotten Password\Base URL' is used to store the URL (Figure 19), which is appended with the validation parameters and GUID. This is then concatenated together with the Start Text and End Text settings to create the body of the email. The CC (if required), From, Mail Format and Subject setting keys are then used to complete the email.

Figure 19:
Setting keys
for Send Pass-
word Reset
Online Mem-
bership



There are two endpoints, as an optional parameter can be passed to this endpoint if an additional parameter needs to be appended to link being created.

Permissions

The **Send Password Reset Online Membership** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Enrolment** and **Finance** methods on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- The user must be a member of the Online Services web config user (os_online) user group.
- **Insert** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the 'Person message' and 'Person message data' classes.
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the 'Online membership' and 'Online membership audit' classes.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/security/onlinemembership/sendpasswordreset/{learners email address}/`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/security/onlinemembership/sendpasswordreset/john.smith@yahoo.co.uk/>

Using this endpoint in conjunction with a Base URL setting such as 'https://qa01.unitetest.zone/WCFWebSite/OnlineServices/password-reset.aspx', would create the following link in the generated email:

<https://qa01.unitetest.zone/WCFWebSite/OnlineServices/password-reset.aspx?olm=john.smith@yahoo.co.uk&rpimg=e8ab95a6-472d-46b5-9eb4-1d1a85210b48>

Endpoint example with optional parameter

The endpoint is as follows:

`https://{host header}/{virtual directory}/security/onlinemembership/sendpasswordreset/{learners email address}/{additional verification key}`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/security/onlinemembership/sendpasswordreset/john.smith@yahoo.co.uk/test1234>

Using this endpoint in conjunction with a Base URL setting such as 'https://qa01.unitetest.zone/WCFWebSite/OnlineServices/password-reset.aspx?key='', would create the following link in the generated email:

<https://qa01.unitetest.zone/WCFWebSite/OnlineServices/password-reset.aspx?key=test1234&olm=john.smith@yahoo.co.uk&rpimg=e8ab95a6-472d-46b5-9eb4-1d1a85210b48>



As you're using an email address you must add a trailer '/' to the URL if an additional key isn't supplied, as it contains a period (.) character.

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

Response

If successful, this endpoint returns 'True' and a record is created in the 'Person message' class with the 'Body1' field containing the link.

Reset Password Online Membership end point

Purpose

The **Reset Password Online Membership** endpoint updates the Online Membership password using data generated by the Send Password Reset endpoint and supplied by the student.

Permissions

The **Reset Password Online Membership** endpoint requires the user specified on the **Web API Users** tab in **Security and Settings** to have the following permissions:

- **Enrolment** and **Finance** methods on the **Settings** form accessed from the **Web API Users** tab in **Security and Settings** (see page 18).
- **Select** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the 'Person message' and 'Person message data' classes.
- **Update** permissions in UNIT-e Database Manager (Users) [UDMUS.exe] for the 'Online membership' and 'Online membership audit' classes.

Endpoint example

The endpoint is as follows:

https://{host header}/{virtual directory}/security/onlinemembership/resetpassword

For example:

<https://webapiinstallsite.unitetest.zone/webapi/security/onlinemembership/resetpassword>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

POST query example



Information can be obtained from the output of the **Send Password Reset Online Membership** endpoint (see page 97).

```
{  
  "email":"john.smith@yahoo.co.uk",  
  "authorisationkey":"8a04761a-d137-46a4-a1fc-2eb2ca6d6179",  
  "dob":"14-Jan-1991",  
  "password":"f00tball"  
}
```

Response

If successful this endpoint returns 'True' and the online membership password is updated.

InvalidateSession method

Purpose

The **InvalidateSession** method must be called at the end of process to invalidate the current session, preventing any more requests.

Endpoint example

The endpoint is as follows:

`https://{host header}/{virtual directory}/InvalidateSession`

For example:

<https://webapiinstallsite.unitetest.zone/WebAPI/InvalidateSession>

Add the GUID returned by the **Login** endpoint (see page 32) to the **Value** field in the request header.

Response

The **InvalidateSession** response is a Boolean (True/False).

Chapter 4:

Testing UNIT-e Web API

You are able to test some of your API endpoints using the **UNITe API Explorer** (see page 105) and the pages within the **Pay360 Test pages** sub-folder in Online Services (see page 115). It's currently possible to test the following:

- *Get class* (see page 107)
- *Find class* (see page 108)
- *Update Property* (see page 109)
- *Export Report* (see page 110)
- *Export Report with Filter* (see page 111)
- *Enrol Student* (see page 112)
- *Enrol Students* (see page 113)
- *(Pay360) CompleteEnrolment.aspx test page* (see page 116)
- *(Pay360) CompleteRegisterCard.aspx test page* (see page 117)
- *(Pay360) TestApi.aspx test page* (see page 118)
- *(Pay360) TestPay360Api.aspx test page* (see page 119)

To facilitate testing, this section gives you the prerequisites (see page 104) and status codes to expect from each endpoint (see page 122).

Prerequisites

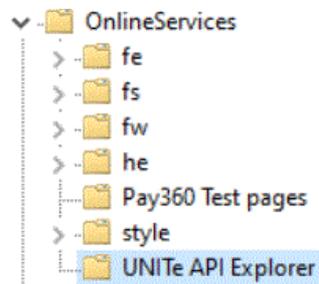
You require the following in order to use the test pages:

- **Secret Key:** The **Secret** value generated for the Web API users. Refer to *The Web API user* (see page 16) for further information.
- **API URL:** This must be https and must have a final forward slash. For example, https://localhost/WebAPI/.
- **API Key (licence authentication key):** The licence authentication key provided by Support Services (e.g. 123456). This is unique and is based on the combination of licence key, version and application name. Version and application name are defined by your institution and form part of all request endpoint URLs.
- **Login URL:** {header}/{virtual directory}/login/{licence key}/{version}/{application name}/{licenceauthentication key}.

UNITe API Explorer

The **UNITe API Explorer** is available in the **Online Services** folder (Figure 20).

Figure 20:
UNITe API
Explorer folder



To use UNITe API Explorer

1. Browse to the **UNITe API Explorer** webpage (Figure 21).

Figure 21:
UNITe API
Explorer

UNITe API Explorer

Secret Key:

API URL:

API Key:

Login URL:

Example URL:

Example Request:

Method	<input type="text"/>	URL	<input type="text"/>	<input type="button" value="Run"/>
---------------	----------------------	------------	----------------------	------------------------------------

Request Body

Endpoints

Response

2. Enter the **Secret Key**, **API URL**, **API key** and **Login URL**.



Refer to *Prerequisites* (see page 104) for further information.

3. Click **Login**. The page then displays the endpoints that may currently be tested (Figure 22).

Figure 22:
UNITe API
Explorer
endpoints

The screenshot shows a list of endpoints in a web application. The list is titled 'Endpoints' and contains the following items: Update Property, Enrol Student, Enrol Students, Get class, Find class, Export report, and Export report w/Filter. Each item is preceded by a small circular icon containing a plus sign.

4. Select an endpoint to view an example appropriate to that endpoint:
 - *Get class* (see below)
 - *Find class* (see page 108)
 - *Update Property* (see page 109)
 - *Export Report* (see page 110)
 - *Export Report with Filter* (see page 111)
 - *Enrol Student* (see page 112)
 - *Enrol Students* (see page 113)

When entering test data you should note the following:

- The names of the fields within the business helper returned in the **Response** don't always match the names of the class properties.
- The order of data in the **Request Body** isn't critical.
- The **Request Body** isn't case-sensitive.

Get class

This GET endpoint returns the business helper record for a specified class and ID.

Figure 23:
UNITe API
Explorer - Get
class

Get class

Example URL:
https://localhost/class/get/student/111000002732365

Example Request:

Method	URL	Run
GET	https://localhost/class/get/student/111000002732365	
Request Body		
<pre> { "id": "1", "type": "UNITe.Business.Helper.Student, UNITe.Business.Helper", "isStudent": true, "studentReference": "S20030001", "previousSystemReference": "", "dateOfFirstContactAsPotentialStudent": null, "dateOfFirstEnrolmentAsActualStudent": null, "currentHomeAddress": "999000021744248", "currentHomeTelephone": "", "currentLocalAddress": "999000021744247", "currentLocalTelephone": "1000010154455", "invoiceAddress": "999000045774975", } </pre>		

URL: Enter in the format shown in the example (e.g. https://{host_header}/{virtual directory}/class/get/{class reference}/{class record id}). In Figure 23 the 'Student' class is specified with a student ID.

Request Body: Not required.

Run: Click to run.

Response: This show the business helper record for the specified class and ID (Figure 24).

Figure 24:
UNITe API
Explorer - Get
class response

Response

```

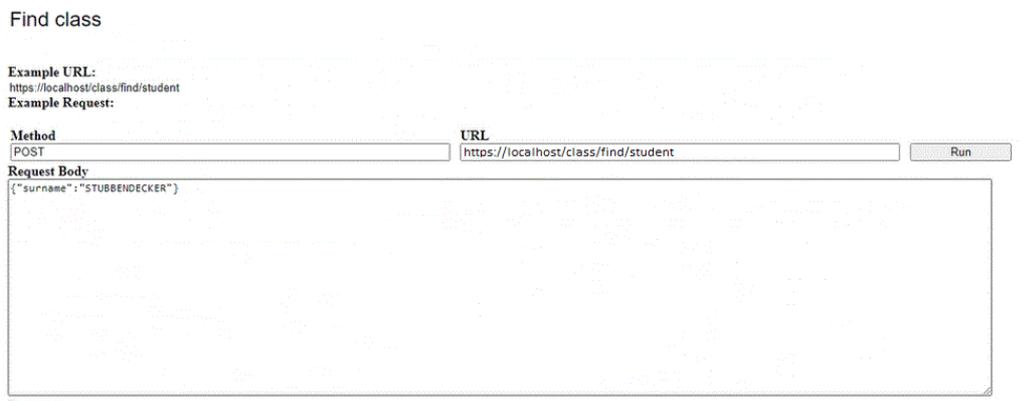
{
  "$id": "1",
  "$type": "UNITe.Business.Helper.Student, UNITe.Business.Helper",
  "IsStudent": true,
  "StudentReference": "S20030001",
  "PreviousSystemReference": "",
  "DateOfFirstContactAsPotentialStudent": null,
  "DateOfFirstEnrolmentAsActualStudent": null,
  "CurrentHomeAddress": "999000021744248",
  "CurrentHomeTelephone": "",
  "CurrentLocalAddress": "999000021744247",
  "CurrentLocalTelephone": "1000010154455",
  "InvoiceAddress": "999000045774975",
}

```

Find class

This POST endpoint goes to the business helper to find and return the requested information.

Figure 25:
UNITe API
Explorer - Find
class



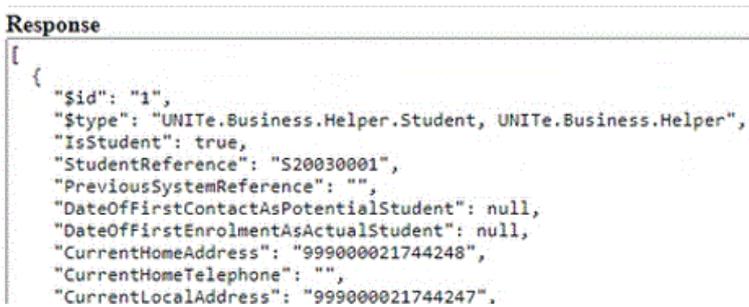
URL: Enter in the format shown in the example (e.g. https://{host header}/{virtual directory}/class/find/{class reference}).

Request Body: Enter the request parameters. The order of data isn't critical and the process isn't case-sensitive. In Figure 25 the example shows a search for any student with a surname of 'Stubbendecker'.

Run: Click to run.

Response: This shows the business helper record for the specified class and **Request Body** (Figure 26).

Figure 26:
UNITe API
Explorer - Find
class response



Update Property

This POST endpoint enables you to update specific properties within a record, based on the class name, record ID and property reference.

Figure 27:
UNITE API
Explorer -
Update
Property

Update Property

Example URL:
https://localhost/class/updateproperty/moduleenrolment

Example Request:
[{"id":"111000046366903","name":"Bob"}]

Method POST **URL** https://localhost/class/updateproperty/student **Run**

Request Body

```
[{"Forenames":"John","Id":"1000010153560"}]
```

URL: Enter in the format shown in the example (e.g. https://{host header}/{virtual directory}/class/updateproperty/{class reference}).

Request Body: Enter the request parameters. The order of data isn't critical and the process isn't case-sensitive. In Figure 27 the example shows a request to update the forename of the student with an ID of '1000010153560' to 'John'.



As this is an array, you can enter more than one POST string.

Run: Click to run.

Response: This shows 'True' if the update is successful (Figure 28).

Figure 28:
UNITE API
Explorer -
Update
Property
response

Response

```
True
```

Export Report

This GET endpoint enables you to extract data from the UNIT-e database by running a UNIT-e Report Generator report using specified run-time parameters. The output is returned in JSON or XML format, depending on the {format} specified in the endpoint.

Figure 29:
UNITe API
Explorer -
Export Report

Export report

Example URL:
https://localhost/report/export/json/student

Example Request:

Method	URL	Run
GET	https://localhost/report/export/json/student	Run
Request Body		

URL: Enter in the format shown in the example (e.g. https://{host header}/{virtual directory}/report/export/{format}/{report reference}). In Figure 29 the example shows the 'Student pick-list report' (student) being used.



The Web API user must have the **Export** method selected on the **Security and Settings** form and associated reports must be specified. Refer to *The Web API user* (see page 16) for further information.

Request Body: Not required.

Run: Click to run.

Response: This shows all records retrieved by the specified report (Figure 30).

Figure 30:
UNITe API
Explorer -
Update Report
response

```

Response
<NewDataSet>
  <student>
    <c1>999000034093707</c1>
    <c2> ?, ?? (01-Jan-1980)</c2>
  </student>
  <student>
    <c1>999000034093712</c1>
    <c2> ?, ?? (01-Jan-1980)</c2>
  </student>
  <student>
    <c1>999000036631909</c1>
    <c2> ?????, ???????? (01-Jan-1980)</c2>
  </student>
  </NewDataSet>
    
```

Export Report with Filter

This GET endpoint enables you to extract specific class properties and values from the UNIT-e database by running a UNIT-e Report Generator report using specified run-time parameters.

Figure 31:
UNITe API
Explorer -
Export Report
with Filter

Export report w/Filter

Example URL:
https://localhost/report/export/json/student/student.forenames.Julia

Example Request:

The screenshot shows a web interface for configuring an API request. It includes a 'Method' dropdown set to 'GET', a 'URL' input field containing 'https://localhost/report/export/json/student/student.surname.smith', and a 'Run' button. Below these is a large empty text area for the 'Request Body'.

URL: Enter in the format shown in the example (e.g. https://{host header}/{virtual directory}/report/export/{format xml/json}/{report reference}/{class},{property},{value}). In Figure 31 the example shows the 'Student pick-list report' (student) being used with the Student class, surname property and a value of 'Smith'.



The Web API user must have the **Export** method selected on the **Security and Settings** form and associated reports must be specified. Refer to *The Web API user* (see page 16) for further information.

Request Body: Not required.

Run: Click to run.

Response: This shows all records retrieved by the specified report, class, property and value (Figure 32).

Figure 32:
UNITe API
Explorer -
Export Report
with Filter
response

```

Response
<NewDataSet>
  <student>
    <c1>999000036631889</c1>
    <c2> Smith, James (01-Jan-1970)</c2>
  </student>
  <student>
    <c1>999000036631894</c1>
    <c2> Smith, James (01-Jan-1970)</c2>
  </student>
  <student>
    <c1>999000105940298</c1>
    <c2> Smith, Jo (</c2>
  </student>
  <student>
    <c1>999000025813629</c1>
    <c2>S000535 Smith, George (02-Feb-1999)</c2>
  </student>

```

Enrol Student

This GET endpoint enrolls a student using the Student Id, Module Id of the Module on which the student wishes to be enrolled and the Fee band reference.

Figure 33:
UNITe API
Explorer -
Enrol Student



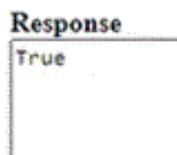
URL: Enter in the format shown in the example (e.g. https://{host header}/{virtual directory}/process/enrolstudent/{student id}/{module id}/{feeband reference}).

Request Body: Not required.

Run: Click to run.

Response: This shows 'True' if the enrolment is successful (Figure 34).

Figure 34:
UNITe API
Explorer -
Enrol Student
response



Enrol Students

The **EnrolStudents** endpoint

This POST endpoint creates multiple enrolments based on the student id, module id and feeband, and optionally updates the status and online reference properties if provided in a single call. As this is an array, multiple parameters can be entered within a single call.



- Only one Status, Online Reference and Fee band may be specified.
- If a module has a mandatory child module, the student will automatically be enrolled on that child module.

Figure 35:
UNITE API
Explorer -
Enrol Students

Enrol Students

Example URL:

`https://localhost/Process/Enrolstudents`

Example Request:

```
[[{"studentid":["111000002732365","111000002698527"],"moduleid":["111000002878072"]}]]
```

Method

POST

URL

`https://localhost/Process/Enrolstudents`

Run

Request Body

```
[[{"StudentID":["1000010153590","1000010153591","1000010153592"],"ModuleID":["1000010150612","1000010150613","1000010150614","1000010150615"],"status":"1","onlinereference":"OLREF4","feeband":"stnd"}]]
```

URL: Enter in the format shown in the example (e.g. `https://{host header}/{virtual directory}/process/enrolstudents`).

Request Body: Enter the request parameters. The order of data isn't critical and the process isn't case-sensitive. In Figure 35 the example shows a request to enrol three students on four modules with a status of '1', an online reference of 'OLREF4' and a fee band of 'stnd'.



As this is an array, you can enter more than one POST string.

Run: Click to run.

Response: This lists each student, the module(s) on which they are enrolled and the enrolment Id, as shown below.

```
[{"StudentID" : "1000010153590"
"Enrolment" : [{"ModuleId" : "1000010150612", "EnrolmentId" : "999000153136010"}]}]
```

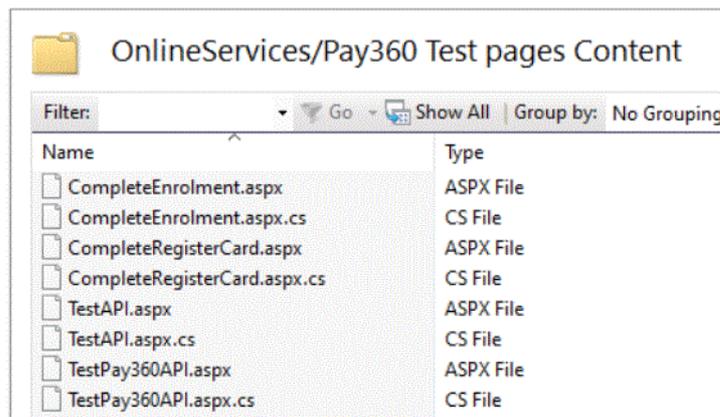
If the chosen module has mandatory child modules, the response also shows those enrolments, as shown below.

```
[
  {"StudentId" : "1000010153590"
  "Enrolment" : [{"ModuleId" : "1000010150612", "EnrolmentId" : "999000153136010"}]}
  {"ModuleId" : "1000010150626", "EnrolmentId" : "999000153136011"},
  {"ModuleId" : "1000010150613", "EnrolmentId" : "999000153136012"},
]
```

Pay360 Test pages

The test pages shown in Figure 36 are provided in Online Services to enable you to test the Pay360 endpoints.

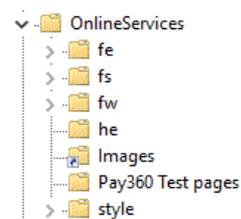
Figure 36:
Pay360 Test
Pages



Name	Type
CompleteEnrolment.aspx	ASPX File
CompleteEnrolment.aspx.cs	CS File
CompleteRegisterCard.aspx	ASPX File
CompleteRegisterCard.aspx.cs	CS File
TestAPI.aspx	ASPX File
TestAPI.aspx.cs	CS File
TestPay360API.aspx	ASPX File
TestPay360API.aspx.cs	CS File

These are within the **Pay360 Test pages** sub-folder shown in Figure 37.

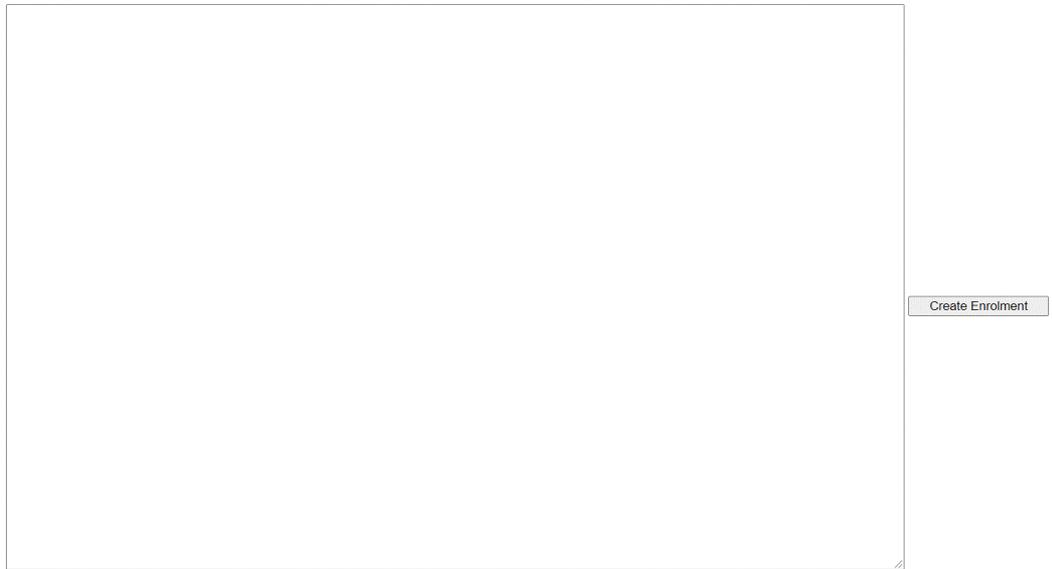
Figure 37:
Pay360 Test
pages folder



- *CompleteEnrolment.aspx test page* (see below)
- *CompleteRegisterCard.aspx test page* (see page 117)
- *TestApi.aspx test page* (see page 118)
- *TestPay360Api.aspx test page* (see page 119)

CompleteEnrolment.aspx test page

Figure 38:
CompleteEnrolment test page



Add JSON input - for example:

```
{
  "Market": "General",
  "EnrolmentFeeDetails": [{
    "EnrolmentFees": [{
      "FeeID": "9999999999999999"
    }]
  }]
  "Source": "Source",
  "FeeBand": "online",
  "StudentID": "9999999999999999"
}
```

Click **Create Enrolment**. The message 'Enrolment created successfully' is returned if the test is successful.

CompleteRegisterCard.aspx test page

Figure 39:
CompleteRegisterCard test page

API : Finance/Pay360CompleteRegisterCard

Request Body	Response Body
<div style="border: 1px solid black; height: 100px;"></div>	<div style="border: 1px solid black; height: 100px;"></div>

[Go Back To Test API Page](#)

Add JSON input - for example:

```
{
  'ScpReference':'br4inlweew136uy6l7rfi7lapzso1j5',
}
```

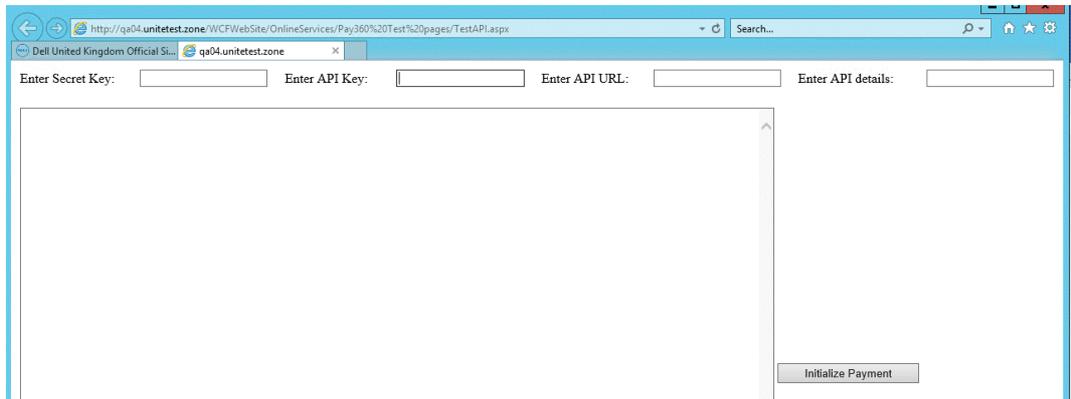
Response

The response is as follows if the test is successful:

```
{
  "Status":"SUCCESS",
  "CardLastFourDigits":"6721",
  "ExpiryDate":"1220"
}
```

TestApi.aspx test page

Figure 40:
Initialise payment (TestApi)



Enter the following:

```
string uri = apiUrl + "/login/" + apiKey + "/" + apiDetails; (e.g. https://webapiinstallsite.unitetest.zone/WebAPI/login/APTS/1.0/Web API Test App 2/542349).
```

Click **Initiate Payment**.

TestPay360Api.aspx test page

Figure 41:
TestPay360Api
test page

Test Pay360 Web API

Secret Key: <input type="text"/>	API Key: <input type="text"/>
API URL: <input type="text" value="example: https://localhost/WebAPI/"/>	API Endpoint: <input type="text" value="--- Select ---"/>
Login URL: <input type="text" value="example: 1.0/Web API/123456"/>	

Request Body

Response Body

Select the API endpoint:

- *Finance/Pay360InitialiseRegisterCard* (see page 119)
- *Finance/Pay360DeleteCard* (see page 120)
- *Finance/Pay360SetDefaultCard* (see page 121)

Finance/Pay360InitialiseRegisterCard

Figure 42:
TestPay360Api
- InitialiseReg-
isterCard

Request Body

```
{
  "StudentID": "123456789",
  "StudentEmail": "xyz@test.com",
  "ReturnUrl": "http://localhost/WCFWebSite/OnlineServices/Pay360 Test
pages/CompleteRegisterCard.aspx"
}
```

When the 'Finance/Pay360InitialiseRegisterCard' option is selected as the **API Endpoint**, the **Request Body** field is automatically populated with the following:

```
{
  "StudentID": "123456789",
  "StudentEmail": "xyz@test.com",
  "ReturnUrl": "http://localhost/WCFWebSite/OnlineServices/Pay360 Test pages/
CompleteRegisterCard.aspx"
```

```
}
```

Amend as required and **Send Request**.

Response

The response is as follows if the test is successful:

```
{
  "RedirectUrl": "https://sbsctest.e-paycapita.com:443/scp/scpcli?ssk=srgj23k63zxhy04zifaf4218ps4no0g",
  "UniqueReference": "9999999999",
  "ScpReference": "br4inlweew136uy6l7rfi7lapzso1j5",
  "Status": "SUCCESS"
}
```

Finance/Pay360DeleteCard

Figure 43:
TestPay360Api
- DeleteCard

Request Body

```
{
  "CardId": "123456789"
}
```

When the 'Finance/Pay360DeleteCard' option is selected as the **API Endpoint**, the **Request Body** field is automatically populated with the following:

```
{
  "CardId": "123456789"
}
```

Amend as required and **Send Request**.

Response

The response is as follows if the test is successful:

```
{
  Status: 'SUCCESS'
}
```

Finance/Pay360SetDefaultCard

Figure 44:
TestPay360Api
- SetDefault-
Card

```
Request Body
{
  "CardId": "123456789"
}
```

When the 'Finance/Pay360SetDefaultCard' option is selected as the **API Endpoint**, the **Request Body** field is automatically populated with the following:

```
{
  "CardId": "123456789"
}
```

Amend as required and **Send Request**.

Response

The response is as follows if the test is successful:

```
{
  'Status' : 'SUCCESS',
}
```

Status codes

- *Get endpoint* (see below)
- *Post endpoint* (see page 122)

Get endpoint

Table 1: Get endpoint status codes

Scenario details	Expected result
Success	Returns 200
No authentication/token	Returns 401 Unauthorised
No permissions to read	Returns 403 Forbidden
No ID supplied as key and No Filter	Either returns all records in collection, all records with paging or validation error with details of filter required – depends on business requirement
Invalid ID (as key)	Returns 404 Not Found
ID supplied in the Filter where the ID does not exist	Returns an empty collection
Valid ID (as key)	Returns 404 Not Found

Post endpoint

Table 2: Post endpoint status codes

Scenario details	Expected result
Success	Returns 200
No authentication/token	Returns 401 Unauthorised
No permissions to write	Returns 403 Forbidden
ID supplied	Returns 400 Bad Request, with appropriate error message
Individually test with each of the mandatory fields not supplied	If fails - returns 400 Bad Request With validation error

Scenario details	Expected result
With multiple mandatory fields not supplied – multiple validation errors	If fails - returns 400 Bad Request With multiple validation errors
Test for mandatory fields not supplied by user input (e.g. learner id, fk reference ids)	If fails - returns 400 Bad Request With validation error (scenario added as this is commonly missed validation)
Other validation errors as specified in requirements (e.g., start date after end date) – test specific business validation	If fails - returns 400 Bad Request With validation errors
Malformed model (incorrect structure)	Returns 400 Bad Request, with appropriate error message

Chapter 5:

Trouble-shooting and tips

This chapter includes the following:

- *Identifying property reference for Insert and Update endpoints* (see page 126)

Identifying property reference for Insert and Update endpoints

Web API uses the business helper reference for all classes and properties. If the class reference doesn't match the business helper reference, use the **Get** endpoint to return a record and then locate the business helper reference in the JSON response, which must be used by other endpoints when updating or inserting records via the Web API.

The business helpers are used by the Web Service/Web API to expose data. The following example (using the personaddress.name field), explains how to reconcile the property reference displayed in Database Manager Class Editor with the business helper reference utilised by the Web API.

The following fields can be updated in the base class (Address) and extended class (Person address).

- <https://hostname/WebAPI/class/update/personaddress>
- <https://hostname/WebAPI/class/updateproperty/personaddress>

For the Address.name property (Figure 45), use the business helper reference, which is AddressDetails. This can be achieved using either the Update endpoint (see page 39) or Update property endpoint (see page 41).

Figure 45:
Address class
showing name
property



Name	Reference	Type	Link class
Id	id	Id	
Start	start	Date	
End	end	Date	
Postcode	reference	Reference	
Address	name	Text	
Country	country	Lookup	Country

You can use a Get endpoint to return an example record (e.g. <https://hostname/WebAPI/class/get/personaddress/10578716>), which displays the JSON below.

```
{
  "$id": "1",
  "$type": "UNITE.Business.Helper.PersonAddress, UNITE.Business.Helper",
  "Type": "Current",
  "Person": "1000010303578",
  "AlternativeCode": "",
  "HESATermTimeAccommodation": "",
  "Start": null,
  "End": null,
  "Postcode": "GL7 1LT",
  "AddressDetails": "5 Mercian Close\r\nWatermoor\r\nCirencester",
}
```

```

"Country": "399",
"Notes": "",
"IsTransferred": false,
"WAUEStatus": "",
"WAUEDateTransferred": null,
"Number": "",
"AddressLine1": "",
"AddressLine2": "",
"AddressLine3": "",
"Town": "",
"County": "",
"UPRN": "",
"USRN": "",
"Easting": "",
"Northing": "",
"IsReturnedMail": false,
"AdultEducationBudgetArea": "",
"AdultEducationBudgetSource": "",
"BFPONumber": null,
"POLAR3Young": "",
"POLAR3Adult": "",
"POLAR4": "",
"DeprivationRank": "",
"DeprivationDecile": "",
"EERName": "",
"Authority": "",
"Ward": "",
"DownloadStatus": "",
"DownloadDate": null,
"Stamp": {
  "$type": "System.Byte[], mscorlib",
  "$value": "AAAAAABjsQ="
},
"ExtendedProperties": null,
"DefaultValues": null,
"Id": "10578716"
}

```

Importing telephone numbers

?? Previously, if you imported a person telephone number with a leading "+", the + was replaced on import with a space. The + sign should be replaced with %2B in the API call to ensure that the + is imported correctly.

Appendices

- *Appendix A: Previous enhancements* (see page 130)

Appendix A: Previous enhancements

The following enhancements were made for previous releases:

- *4.52 enhancements* (see below)
- *4.51 enhancements* (see page 131)
- *4.49 enhancements* (see page 131)

4.52 enhancements

- *EnrolStudents endpoint* (see page 66)
 - The EnrolStudents endpoint now creates multiple enrolments based on the student id, module id and feeband, and optionally updates the status and online reference properties if provided in a single call.
 - If a module has mandatory child modules, the student will be automatically enrolled on these modules.
 - As this is an array, multiple parameters can be entered within a single call.
- *Register week pattern updates from register marks endpoint* (see page 69)

Further information has been added to this section to explain the relationship between the "id" and "type".
- *UNITE API Explorer* (see page 105)

This new page is provided to enable you to test the following API endpoints:

- Get
- Find
- UpdateProperty (with both a single value and an array)
- EnrolStudent
- EnrolStudents
- Export Report
- Export Report with Filter

4.51 enhancements

- *Insert endpoint* (see page 48)

A note has been added to this section to explain that if you're dealing with a write-back to the 'Person message' class, any email text supplied in the 'Body 1' property that exceeds 2,000 characters is automatically distributed to properties 'Body 2' to 'Body 6' (provided no values have been supplied in these properties) and the 'Person message data' class.

- *Pay360 Test pages* (see page 115)

This section has been updated to include the following:

- *CompleteEnrolment.aspx test page* (see page 116)
- *CompleteRegisterCard.aspx test page* (see page 117)
- *TestPay360Api.aspx test page* (see page 119)

- *Chapter 5: Trouble-shooting and tips* (see page 125)

This new chapter has been added to include trouble-shooting suggestions and tips.

4.49 enhancements

- *Building user class assemblies* (see page 25)

This new section has been added to explain how to build user class assemblies. This process must be carried out if you wish to run Web API processes, IICL processes or both against user classes.

Glossary

API (Application Programming Interface)

An interface that has a set of functions (e.g. subroutine definitions, protocols, and tools) that allow programmers to access data within an application or specific features of an application. For example, an API would provide a way for third-party software to access your UNIT-e Student records.

GUID (Globally Unique Identifier)

A unique identifier generated by an algorithm. Also known as a UUID (Universally Unique Identifier).

https (Hypertext Transfer Protocol Secure)

A set of rules for transferring files via the Internet, https is an extension of the Hypertext Transfer Protocol (http) that is used to provide secure communication over a computer network.

POST

POST is a request endpoint supported by http. This endpoint requests that a web server accepts the data enclosed in the body of the request message. It's commonly used when uploading a file or when submitting a completed web form.

REST (Representational State Transfer)

An architectural style for networked hypermedia applications. It's primarily used to build web services that are lightweight, maintainable and scalable. UNIT-e API is a REST API.

Web API: An API that can be accessed over the web using https.

Index



If you are unable to find the information you require, the search facility in Adobe Acrobat enables you to search individual PDFs or the entire UNIT-e documentation catalogue. Refer to the *About UNIT-e documentation* section in the *Working with UNIT-e* handbook ([WWU.pdf](#)) for further details.

A

API

- communicating with, 31
- installation, 19
- referencing, 24
- testing, 103
 - prerequisites, 104
 - status codes by endpoint, 122

E

endpoints

- Create, 46
- Create Online Membership, 91
- Delete, 37
- Enrol Student, 64
- Enrol Students, 66
- Export, 52
- Export Schema, 61
- Export with parameters, 55
- Export with parameters using special characters, 57
- Find, 44
- Get, 35
- Insert, 48
- Lock, 50
- Pay360CompleteEnrolment endpoint
 - single enrolment and single fee, 77
- Pay360CompleteRegisterCard, 84
- Pay360DeleteCard, 86
- Pay360InitialisePayment endpoint, 72
- Pay360InitialiseRegisterCard, 88
- Pay360SetDefaultCard, 82
- Register week pattern updates, 69
- Report
 - Import, 59
- Reset Password Online Membership, 100
- Send Password Reset Online Membership, 97
- testing, 23

Update, 39

Update Online Membership, 95

Update property, 41

Validate Online Membership, 93

I

installation, 19

L

LDAP, 17

licence requirements, 14

M

methods, 29

Class, 33

Finance, 71

InvalidateSession, 102

Login, 32

Process, 63

Report, 51

Security, 90

O

online membership *see 'methods - Security'*
overview, 9

P

Pay360, 71

- test pages, 115
 - CompleteEnrolment, 116
 - CompleteRegisterCard, 117
 - TestApi, 118
 - TestPay360Api, 119

W

- web.config file adjustment, 20

S

- setup requirements, 13
 - Capita.lic file, 22
 - installation, 19
 - licence, 14
 - testing method calls, 23, 122
 - URL endpoint, 24
 - user access, 15
 - web.config file adjustment, 20
- status codes, 122

T

- testing
 - Enrol Student, 112, 113
 - Export Report, 110
 - Export Report with Filter, 111
 - Find class, 108
 - Get class, 107
 - Pay360
 - CompleteEnrolment, 116
 - CompleteRegisterCard, 117
 - Finance/Pay360DeleteCard, 120
 - Finance/Pay360InitialiseRegisterCard, 119
 - Finance/Pay360SetDefaultCard, 121
 - TestAPI, 118
 - TestPay360Api, 119
 - Pay360 test pages, 115
 - prerequisites, 104
 - status codes by method, 122
 - test database connection, 27
 - UNITe API Explorer, 105
 - Update Property, 109
- trouble-shooting and tips, 125
 - property reference for insert & update endpoints, 126

U

- URL endpoint, 24
- user access, 15
 - web API user, 16
 - web config file user, 15
- user class assemblies, 25
- user group, 15

List of figures

Figure 1:	The Web API process	10
Figure 2:	Security and Settings - Web API Users.....	17
Figure 3:	Security and Settings - secret key.....	18
Figure 4:	Security and Settings - Settings button	18
Figure 5:	Security and Settings - Settings.....	18
Figure 6:	Web API Installation.....	19
Figure 7:	web.config file	20
Figure 8:	Web API Application Settings	20
Figure 9:	Capita.lic file in WebAPI folder	22
Figure 10:	Build user class assemblies	25
Figure 11:	UNITE.Database.dll and UNITE.Database.BusinessClasses.dll.....	26
Figure 12:	GUID added to request header.....	32
Figure 13:	API Delete - response unsuccessful.....	38
Figure 14:	Property reference.....	42
Figure 15:	Export with parameters using special characters - report details	58
Figure 16:	API Export Schema response.....	62
Figure 17:	API Create Online Membership response.....	92
Figure 18:	API Validate Online Membership response.....	94
Figure 19:	Setting keys for Send Password Reset Online Membership.....	97
Figure 20:	UNITE API Explorer folder.....	105
Figure 21:	UNITE API Explorer	105
Figure 22:	UNITE API Explorer endpoints	106
Figure 23:	UNITE API Explorer - Get class.....	107
Figure 24:	UNITE API Explorer - Get class response.....	107
Figure 25:	UNITE API Explorer - Find class.....	108
Figure 26:	UNITE API Explorer - Find class response.....	108
Figure 27:	UNITE API Explorer - Update Property	109
Figure 28:	UNITE API Explorer - Update Property response	109
Figure 29:	UNITE API Explorer -Export Report.....	110
Figure 30:	UNITE API Explorer - Update Report response.....	110
Figure 31:	UNITE API Explorer - Export Report with Filter	111
Figure 32:	UNITE API Explorer - Export Report with Filter response	111
Figure 33:	UNITE API Explorer - Enrol Student	112

Figure 34:	UNITe API Explorer - Enrol Student response	112
Figure 35:	UNITe API Explorer - Enrol Students	113
Figure 36:	Pay360 Test Pages	115
Figure 37:	Pay360 Test pages folder	115
Figure 38:	CompleteEnrolment test page.....	116
Figure 39:	CompleteRegisterCard test page	117
Figure 40:	Initialise payment (TestApi).....	118
Figure 41:	TestPay360Api test page.....	119
Figure 42:	TestPay360Api - InitialiseRegisterCard	119
Figure 43:	TestPay360Api - DeleteCard	120
Figure 44:	TestPay360Api - SetDefaultCard	121
Figure 45:	Address class showing name property	126