INTRODUCTION TO ENFIX

Welcome to Enfix, an embeddable SCORM-conformant learning management system!

Enfix offers two modes of operation for integration: a traditional environment where users log directly into an LMS interface through their web browser directly, and an embedded environment designed to be integrated with an existing web experience that users are already familiar with.

Enfix is composed of two separate systems, a LRS and an LMS UI.

The Learning Record Store (LRS) is the data storage layer for the entire system, isolating all the tracking of data independently from the display of the learning experience. This allows for greater flexibility in embedding the learning experience in other systems. The LRS exposes all data via a standard HTTP REST API, making it operable as a completely independent entity for other systems to interact with the data from the learning experiences.

The LMS UI provides a traditional learning management environment for users. This system is customizable and themeable to look like your existing application experience, and provides all the necessary components for the serving and playing of SCORM-conformant courseware.

INSTALLATION

Here's the installation instructions for Enfix.

Assumptions:

1. Ubuntu 13.04
2. We will be using nginx as the web server, and UWSGI as the adapter. Apache and modwsgi are also possible, but not described here.

```
apt-get install nginx uwsgi python2.7 python2.7-dev python-virtualenv
mongodb uwsgi-plugin-python \
    libjpeg62 libpng3 libjpeg62-dev libpng12-dev zlib1g-dev
libfreetype6 libfreetype6-dev
cd /home
mkdir -p sites/venv/
cd sites
# Unzip Enfix packages here.

virtualenv venv/gglms_api
source venv/gglms_api/bin/activate
```

```
cd gglms_api
pip install -r requirements.txt
# Make a prod.py config.
python setup_site.py <domain_prefix>
cp wsgi.wsgi wsgi_config.py
# Create /etc/nginx/sites-available/gglms_api
ln -s /etc/nginx/sites-available/gglms_api
/etc/nginx/sites-enabled/gglms_api
# Create /etc/uwsgi/apps-available/gglms_api.xml
ln -s /etc/uwsgi/apps-available/gglms_api.xml
/etc/uwsgi/apps-enabled/gglms_api.xml
service uwsgi restart
service nginx restart

cd ..
virtualenv venv/gglms_ui
source venv/gglms_ui/bin/activate
cd gglms_ui
pip install -r requirements.txt
# Make a settings_local.py config.
python setup_site.py <domain_prefix>
cp wsgi.wsgi wsgi_config.py
# Create /etc/nginx/sites-available/gglms_ui
ln -s /etc/nginx/sites-available/gglms_ui
/etc/nginx/sites-enabled/gglms_ui
# Create /etc/uwsgi/apps-available/gglms_ui.xml
ln -s /etc/uwsgi/apps-available/gglms_ui.xml
/etc/uwsgi/apps-enabled/gglms_ui.xml
service uwsgi restart
service nginx restart
```

/etc/nginx/sites-available/gglms_api:

```
server {
    listen          80;
    server_name     <prefix>.api.enfixlp.com;

    location / {
        uwsgi_pass      unix:///var/run/uwsgi/app/gglms_api/socket;
        include         uwsgi_params;
        uwsgi_param     UWSGI_SCHEME $scheme;
        uwsgi_param     SERVER_SOFTWARE    nginx/$nginx_version;

    }

    location /static {
        root   /home/sites/gglms_api/;
    }

}
```

/etc/uwsgi/apps-available/gglms_api.xml:

```
<uwsgi>
    <plugins>python</plugins>
    <socket>/var/run/uwsgi/app/gglms_api/socket</socket>
    <pythonpath>/home/sites/gglms_api/</pythonpath>
    <virtualenv>/home/sites/venv/gglms_api/</virtualenv>
    <module>wsgi_config:application</module>
    <master/>
    <processes>4</processes>
    <harakiri>60</harakiri>
    <reload-mercy>8</reload-mercy>
    <cpu-affinity>1</cpu-affinity>
    <max-requests>2000</max-requests>
    <limit-as>256</limit-as>
    <reload-on-as>192</reload-on-as>
    <reload-on-rss>128</reload-on-rss>
    <no-orphans/>
    <vacuum/>
</uwsgi>
```

/etc/nginx/sites-available/gglms_ui:

```
server {
    listen          80;
    server_name     <prefix>.ui.enfixlp.com;

    location / {
        uwsgi_pass      unix:///var/run/uwsgi/app/gglms_ui/socket;
        include         uwsgi_params;
        uwsgi_param     UWSGI_SCHEME $scheme;
        uwsgi_param     SERVER_SOFTWARE    nginx/$nginx_version;

    }
}
```

/etc/uwsgi/apps-available/gglms_ui.xml:

```
<uwsgi>
    <plugins>python</plugins>
    <socket>/var/run/uwsgi/app/gglms_ui/socket</socket>
    <pythonpath>/home/sites/gglms_ui/</pythonpath>
    <virtualenv>/home/sites/venv/gglms_ui/</virtualenv>
    <module>wsgi:application</module>
    <master/>
    <processes>4</processes>
    <harakiri>60</harakiri>
    <reload-mercy>8</reload-mercy>
    <cpu-affinity>1</cpu-affinity>
    <max-requests>2000</max-requests>
    <limit-as>256</limit-as>
    <reload-on-as>192</reload-on-as>
    <reload-on-rss>128</reload-on-rss>
    <no-orphans/>
    <vacuum/>
```

```
</uwsgi>
```

QUICKSTART

Assumptions:

1. You have the following credentials (which should have been generated on install or e-mailed to you if you're using the cloud version):
    1. Your admin username
    2. Your admin password
    3. An app token
2. You have the following URLs:
    1. The UI URL. In the cloud version, it looks like: https://<yournamehere>.enfixlp.com/api/v1/
    2. The LRS URL. In the cloud version, it looks like: https://<yournamehere>.enfixlp.com/api/v1/
3. You have cURL or an equivalent HTTP fetching tool to run the commands. Some recommendations:
    1. Postman
    2. cURL
    3. httpie

Generating a user token

Every request requires two tokens: an app token and a user token. App tokens are meant to live long-term, and only expired when desired. A user token is intended to last for about 24 hours (configurable in full installations), and provides the context for those requests.

Token requests are one of the few traditional POSTs in the system, in case you need to make this request through a traditional browser form, and accepts a username and password. It looks like the following:

```
curl -X POST -H "App-Token: [APP_TOKEN]" -d
"username=[username]&password=[password]" \
    https://<yournamehere>.enfixlp.com/api/v1/token/
```

Your token return will look like:

```
{"token": "8031ac4ce22c"}
```

Discovering the available endpoints

All of the API endpoints not related to token generation use a standard HTTP REST mechanism, and use the following standard HTTP verbs:

1. GET - Retrieving a single or list of objects.
2. POST - Creating a single object.
3. PUT - Completely replacing a single object.
4. PATCH - Updating a portion of a single object.
5. DELETE - Removing a single object.

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]" \
     -H "Content-Type: application/json" \
     https://<yournamehere>.enfixlp.com/api/v1/
```

Output should look similar to the following:


```
{
    "_links": {
        "child": [
            {
                "href":
"https://<yournamehere>.enfixlp.com/api/v1/users/",
                "title": "Users"
            },
            {
                "href":
"https://<yournamehere>.enfixlp.com/api/v1/groups/",
                "title": "Groups"
            },
            {
                "href":
"https://<yournamehere>.enfixlp.com/api/v1/courses/",
                "title": "Courses"
            },
            {
                "href":
"https://<yournamehere>.enfixlp.com/api/v1/activity_sessions/",
                "title": "Activity Sessions"
            },
            {
                "href":
"https://<yournamehere>.enfixlp.com/api/v1/learning_paths/",
                "title": "Learning Paths"
            },
            {
                "href":
"https://<yournamehere>.enfixlp.com/api/v1/roles/",
                "title": "Roles"
            },
            {
                "href":
"https://<yournamehere>.enfixlp.com/api/v1/settings/",
                "title": "Settings"
            }
```

```
        ],
        "views": [
            {
                "href":
"https://<yournamehere>.enfixlp.com/api/v1/views/user/<string:user_id>/
courses_enrolled_in/",
                "title": "User's Enrolled Courses"
            },
            {
                "href":
"https://<yournamehere>.enfixlp.com/api/v1/views/user/<string:user_id>/
courses_by_status/",
                "title": "User's Courses By Status"
            },
            {
                "href":
"https://<yournamehere>.enfixlp.com/api/v1/views/course/<string:course_
id>/users_enrolled_in/",
                "title": "Course's Enrolled Users"
            }
        ]
    }
}
```

Each of these describes a valid endpoint to go looking for more data. These endpoints are also visible via the API Navigator.

Retrieving a list of users

An API endpoint that provides a list will always end in a trailing slash. For example, this request shows a list of users:

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]" \
     -H "Content-Type: application/json" \
     https://<yournamehere>.enfixlp.com/api/v1/users/
```

The status code of your request should return a 200. You should receive output similar to the following (pretty-printed for readability via “python -m json.tool”):

```
[
    {
        "_id": "514143a7bd7a84210c88ca1d",
        "active": true,
        "date_added": "2013-03-17 23:52:26",
        "date_updated": "2013-03-13 23:27:35",
        "email": "bob@bob.com",
        "name": "Bob",
        "roles": [
            "51298e25835fc685850316d2"
        ],
        "username": "bob"
```

```
    },
    {
        "_id": "51b1e78dbd7a844089b652a9",
        "active": true,
        "date_updated": "2013-10-16 16:48:54",
        "email": "gav15@gav15.com",
        "name": "George Vilches",
        "roles": [],
        "username": "gav15"
    },
    {
        "_id": "51b1e833bd7a844089b652aa",
        "active": true,
        "date_updated": "2013-08-25 12:09:56",
        "email": "a@e.com",
        "name": "George Vilches2",
        "roles": [
            "51298e25835fc685850316d2"
        ],
        "username": "gav163"
    }
]
```

If you do not receive either the correct status code or a JSON body that looks similar to the above, you will want to review the Debugging API Error Codes documentation.

Retrieving a single user

An API endpoint that provides a single object will never end in a trailing slash, and will generally have a 24-charcter string as the identifier (usually stored in the "_id" field.) The request to fetch a single user from our list above looks like:

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]" \
     -H "Content-Type: application/json" \

https://<yournamehere>.enfixlp.com/api/v1/users/51b1e833bd7a844089b652a
a
```

The user object should match the object as fetched from the list view, without the additional objects:

```
{
    "_id": "51b1e833bd7a844089b652aa",
    "active": true,
    "date_updated": "2013-08-25 12:09:56",
    "email": "a@e.com",
    "name": "George Vilches2",
    "roles": [
        "51298e25835fc685850316d2"
    ],
    "username": "gav163"
```

```
}
```

Creating a user via the API

All user creations are done via the POST verb, and the POST body is expected to be a JSON object of all the fields describing that user. Creations always go to the main endpoint and not to a specific entity, and the endpoint must always end in a URL. An example:

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]" -H
"Content-Type: application/json" \
    -d '{"username": "joe", "password": "asdfasdfasdf", "name": "Test
User", "email": "joe123@joe.com"}' \
    "https://<yournamehere>.enfixlp.com/api/v1/users/"
```

If successful, you will receive a HTTP 204 status, and the response body will contain something similar to:

```
{
    "resource_url":
"https://<yournamehere>.enfixlp.com/api/v1/users/51b1e78dbd7a844089b652
a9/",
    "_id": "51b1e78dbd7a844089b652a9"
}
```

You can use the get single command from above to fetch the user from the resource_url listed here.

Updating an object

Updating a user can be done in two ways. Both methods require you use the endpoint for a single object.

First, with PUT, you have to send the whole object, and it will completely replace whatever exists:

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]" -H
"Content-Type: application/json" \
    -d '{"username": "joe", "password": "asdfasdfasdf", "name": "Test
User 2", "email": "joe124@joe.com"}' \

"https://<yournamehere>.enfixlp.com/api/v1/users/51b1e78dbd7a844089b652
a9"
```

If successful, the body will have no content, and the status code will be 204.

If you just want to modify one or a few fields (in our opinion the common use case), you would use PATCH instead. PATCH expects for the data block a JSON object containing only the fields that are changing, and will overlay those onto the base object:

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]" -H
"Content-Type: application/json" \
```

```
    -d '{"username": "joe", "password": "asdfasdfasdf", "name": "Test
User 2", "email": "joe124@joe.com"}' \

"https://<yournamehere>.enfixlp.com/api/v1/users/51b1e78dbd7a844089b652
a9"
```

If successful, the body will have no content, and the status code will be 204.

Filtering a list

For filtering:

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]" -H
"Content-Type: application/json" \

"https://<yournamehere>.enfixlp.com/api/v1/users/?q_username__startswit
h=gav"
```

Returns:

```
[
    {
        "_id": "51b1e78dbd7a844089b652a9",
        "active": true,
        "date_updated": "2013-10-16 16:48:54",
        "email": "gav15@gav15.com",
        "name": "George Vilches",
        "roles": [],
        "username": "gav15"
    },
    {
        "_id": "51b1e833bd7a844089b652aa",
        "active": true,
        "date_updated": "2013-08-25 12:09:56",
        "email": "a@e.com",
        "name": "George Vilches2",
        "roles": [
            "51298e25835fc685850316d2"
        ],
        "username": "gav163"
    }
]
```

It's a URL param, q_<field>=whatever (exact match), or q_<field>__<filtertype>=whatever.

Supported filter types are:

1. exact - Matches only the exact phrase specified. Implicit if you don't specify a filter type.

2. startswith - Acts like “whatever”% or “whatever”.* in SQL/Regex notation, respectively. Matches anything that starts with the specified string.

Embedding just the player

If you want to embed just the player within your existing site, first follow the instructions to create a token for that user via the API. Then, create an iframe to your content with the following URL:

```
https://<yournamehere>.enfixlp.com/api/v1/ssla_tt/<course_id>/?username
=<username>&token=<to
```

DEBUGGING API ERROR CODES

Highlights of possible errors and how to resolve them.

HTTP Status Code 400 - No JSON object could be decoded

If you receive a 400 with just the string “No JSON object could be decoded”, the data block of your POST/PUT/PATCH is invalid JSON. Run it through a JSON parser to identify what is incorrectly encoded in your packet and submit again.

HTTP Status Code 400 - {“errors”: …} block

If you receive a 400 with a JSON document containing only an “errors” key, this almost always indicates a validation error in the submitted data. The errors key contains a hash of keys named after the various fields, and an error message appropriate to that field for the value. The “_” key or “” (empty) key represents errors that do not apply to a specific field or other general validation issues.

An example:

```
{
    "errors": {
        "": "The username \"joe\" already exists, please select
another."
    }
}
```

AUTHENTICATION

Every request requires two tokens: an app token and a user token. App tokens are meant to live long-term, and only expired when desired. A user token is intended to last for about 24 hours (configurable in full installations), and provides the context for those requests.

Basic token (login) requests are one of the few traditional POSTs in the system, in case you need to make this request through a traditional browser form, and accepts a username and password. It looks like the following:

```
curl -X POST -H "App-Token: [APP_TOKEN]" -d
"username=[username]&password=[password]" \
    https://<yournamehere>.enfixlp.com/api/v1/token/
```

Your token return will look like:

```
{"token": "8031ac4ce22c"}
```

Creating Application Tokens

Application tokens are managed in the same fashion as the other REST endpoints, with a few differences:

1. *PUT is not allowed.*
2. *You cannot specify your own token string, you can only specify the expiration.*

There is no maximum expiration date. By default, application tokens expire about 200 years from now (configurable in full installations). That should be plenty of time for you to refresh your tokens!

To create a new application token:

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]" -H
"Content-Type: application/json" \
    -d '{"seconds_until_expire": 86400}' \
    "https://<yournamehere>.enfixlp.com/api/v1/app_tokens/"
```

If you successfully create an app token, you'll receive the following response:

```
{"app_token": "5aa59576866e4167b26c8430fd4e4d29"}
```

Authenticating On Behalf Of Another User (aka OBO Tokens)

If you want to be able to generate tokens on behalf of (OBO) another user, there's a "tokens" endpoint that works very similarly to the Application Tokens endpoint above. It has the same limitations. Default expiration is a day (configurable in full installations).

To create an OBO token:

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]" -H
"Content-Type: application/json" \
    -d '{"seconds_until_expire": 86400, "user":
"52c5fc24a64c9efc0f253535"}' \
    "https://<yournamehere>.enfixlp.com/api/v1/tokens/"
```

The user field is the ID of the user you're updating.

If you successfully create an OBO token, you'll receive the following response:

```
{"token": "e3bb632cd0d441a2adc9fc47507cf3fe"}
```

If you specify an invalid user ID, you should receive the following error:

```
{
    "errors": {
        "user": "One or more IDs are not formatted correctly."
    }
}
```

Changing a Token's Expiration

Both tokens and app tokens can have their expiration date changed or set in the past as a way to immediatley invalidate without necessarily deleting all record of the token. It's done as a PATCH operation just like the other PATCH operations elsewhere in the API, and is set via `seconds_until_expire`. This value is always from the current moment in time, so setting it to -1 or less will immediately expire the token.

Deleting a Token

If you must remove a token or app token, the standard DELETE verb works on it. The ID is the same as the token itself, so specifying that as part of the token/app token URL path will be sufficient.

COURSE ENROLLMENT

There are four ways to make a course available to a student:

1. *Set the course's "Requires Enrollment" property to false. (Of course, this makes the course available to all users of the LMS, so make sure this is what you want!)*
2. *Being added directly as an enrolled user.*
3. *Being added to a group that is an enrolled group.*
4. *Being added to a group that is a subgroup of an enrolled group, and having the "include subgroups" option selected for the course.*

Enrolling a User Directly

To enroll a user, you'll want to fetch the course, and then update the "enrolled_users" property array with your new user and PATCH it back into the course. Users should be added to the array by their system ID, not their username:

```
curl -X GET -H "App-Token: guddiclealCoph" -H "Token:
4599b52e75b4482ca11969ec2aada5f9" \
    -H "Content-Type: application/json" \

"https://<yournamehere>.enfixlp.com/api/v1/courses/5144e53dbd7a84023457
4ee4"
```

Return looks something like:

```
{
    "_id": "5144e53dbd7a840234574ee4",
    "active": true,
    "date_added": "2013-03-17 23:52:40",
    "date_updated": "2013-03-16 17:43:20",
    "description": "",
    "enrolled_groups": [
        "51413a49bd7a841b9ca9528b",
        "514240aabd7a8474c26de51a"
    ],
    "enrolled_users": [],
    "title": "Course 3"
}
```

Then you would PATCH the user list with the new user ID:

```
curl -X PATCH -H "App-Token: guddiclealCoph" -H "Token:
4599b52e75b4482ca11969ec2aada5f9" \
```

```
    -H "Content-Type: application/json" -d '{"enrolled_groups":
["52d684c8bd7a8442e3365556"]}' \

"https://<yournamehere>.enfixlp.com/api/v1/courses/5144e53dbd7a84023457
4ee4"
```

Enrolling a User Via a Group

To enroll a user via a group, you'll want to fetch the group and add the user to the "members" of the group. From there, you'll want to fetch the course object as above and update the "enrolled_groups" property:

```
curl -X PATCH -H "App-Token: guddiclealCoph" -H "Token:
4599b52e75b4482ca11969ec2aada5f9" \
    -H "Content-Type: application/json" \
    -d '{"enrolled_groups": ["52d684c8bd7a8442e3365556"]}' \

"https://<yournamehere>.enfixlp.com/api/v1/courses/5144e53dbd7a84023457
4ee4"
```

Enrolling a User Via a Sub-Group

To enroll a user via a sub-group, you'll want to follow all the steps above in Enrolling a User Via a Group. You'll want to add one additional field into a PATCH: "include_subgroups", with the parent group ID that contains all the groups you want enrolled (make sure this group is also in the "enrolled_groups":

```
curl -X PATCH -H "App-Token: guddiclealCoph" -H "Token:
4599b52e75b4482ca11969ec2aada5f9" \
    -H "Content-Type: application/json" \
    -d '{"enrolled_groups": ["52d684c8bd7a8442e3365556"],
"include_subgroups": ["52d684c8bd7a8442e3365556"]}' \

"https://<yournamehere>.enfixlp.com/api/v1/courses/5144e53dbd7a84023457
4ee4"
```

CERTIFICATES

A common courseware requirement is to produce a stamped certificate for students that have completed material. This generally needs to include custom information about the student and the material being taken, dates and so on. Described below is a system for defining a certificate-creating mechanism that can be stored for the user in your application.

An Example Certificate Data Structure

The following demonstrates many of the potential "stampable" types when building a certificate. POST this to your API endpoint with the following cURL, and then save the ID for the next step.

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]" -H
"Content-Type: application/json" -d '
{
    "cert_type": "pdf",
    "output_format": "A4",
    "orientation": "landscape",
    "stampables": [
        {
            "height": null,
            "img": "https://www.google.com/images/srpr/logo11w.png",
            "stamp_type": "image",
            "width": 40,
            "x": null,
            "x_align": "center",
            "x_align_shift": null,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 5
        },
        {
            "height": null,
            "stamp_type": "text",
            "text": "ACCREDITATION COUNCIL FOR",
            "width": null,
            "x": null,
            "x_align": "center",
            "x_align_shift": null,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 31
        },
        {
            "height": null,
            "stamp_type": "text",
            "text": "CONTINUING MEDICAL EDUCATION",
            "width": null,
            "x": null,
            "x_align": "center",
            "x_align_shift": null,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 35
        },
        {
            "font": "Times",
```

```
            "font_size": 16,
            "font_style": "I",
            "height": null,
            "stamp_type": "text",
            "text": "This is to certify that",
            "width": null,
            "x": null,
            "x_align": "center",
            "x_align_shift": null,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 44
        },
        {
            "height": null,
            "stamp_type": "text",
            "text": "{{NAME}}, {{DEGREE}}",
            "width": null,
            "x": null,
            "x_align": "center",
            "x_align_shift": null,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 51
        },
        {
            "font": "Times",
            "font_size": 14,
            "font_style": "",
            "height": null,
            "stamp_type": "text",
            "text": "has participated in the enduring material titled
{{ACTIVITY}} on {{CURRENT_DATE}}",
            "width": null,
            "x": null,
            "x_align": "center",
            "x_align_shift": null,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 58
        },
        {
            "font": "Times",
            "font_size": 14,
            "font_style": "I",
```

```
            "height": null,
            "stamp_type": "text",
            "text": "This meeting has been accredited by the Council
for Medical Education and is designated for a maximum of 2 credits",
            "width": null,
            "x": null,
            "x_align": "center",
            "x_align_shift": null,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 66
        },
        {
            "font": "Times",
            "font_size": 10,
            "font_style": "",
            "height": null,
            "stamp_type": "text",
            "text": "Each participant should claim only those hours of
credit that have actually been spent in the educational activity.",
            "width": null,
            "x": null,
            "x_align": "center",
            "x_align_shift": null,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 76
        },
        {
            "font": "Times",
            "font_size": 10,
            "font_style": "",
            "height": null,
            "stamp_type": "text",
            "text": "_____",
            "width": null,
            "x": null,
            "x_align": "center",
            "x_align_shift": 90,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 89
        },
        {
            "font": "Times",
```

```
            "font_size": 10,
            "font_style": "",
            "height": null,
            "stamp_type": "text",
            "text": "Doctor Bob, MD",
            "width": null,
            "x": null,
            "x_align": "center",
            "x_align_shift": 90,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 92
        },
        {
            "font": "Times",
            "font_size": 10,
            "font_style": "",
            "height": null,
            "stamp_type": "text",
            "text": "Course Director",
            "width": null,
            "x": null,
            "x_align": "center",
            "x_align_shift": 90,
            "x_percent": null,
            "y": null,
            "y_align": null,
            "y_align_shift": null,
            "y_percent": 94
        }
    ]
}' https://<yoursitehere>.enfixlp.com/api/v1/certificates/
```

You'll see the standard POST response:

```
{"resource_url":
"https://<yoursitehere>.enfixlp.com/api/v1/certificates/52e9d26fa64c9e8
3498dea7d", "_id": "52e9d26fa64c9e83498dea7d"}
```

Fetching a Certificate

To fetch the base certificate, you'll want to use the ID from above in the following endpoint:

```
https://<yoursitehere>.enfixlp.com/api/v1/certificate_generator/52e9d26
fa64c9e83498dea7d.pdf
```

Certificates can be customized via the GET parameters. In the “text” or “img” key on a stampable, anything specified inside double curly braces (`{{...}}`) is treated as a variable name to populate from the GET. For example:

```
https://<yoursitehere>.enfixlp.com/api/v1/certificate_generator/52e9d26fa64c9e83498dea7d.pdf?NAME=John%20Thomas&DEGREE=M.D.
```

Currently Supported Output Formats

At the moment, only PDF is supported. Specify a `cert_type` of `pdf` to provide this.

Paper Types

`output_format` can be one of: `A3, A4, A5, Letter, Legal`. Defaults to `Letter`.

`orientation` is either `portrait` or `landscape`. Defaults to `landscape`.

How Stampable Positioning Works

For each of the x and y coordinate, there are 4 potential positioning factors:

1. *[x/y]*. This places the item directly at that coordinate. Be careful that the coordinates are appropriate to the space that you’re working in. PDF coordinate space is not the same as traditional image coordinate space.
2. *[x/y]*_percent. Given the width of the entire document, places the left/top edge of the element at that percentage of the document’s dimension appropriate to the coordinate.
3. *[x/y]*_align. This places the item in the document with the given alignment. Valid options are:
    - *left*: Fully left-aligned, very left edge of the document.
    - *right*: Fully right-aligned, very right edge of the document.
    - *center*: Centered, measured string put directly in center of document, balanced equal lengths on both sides of center.
4. *[x/y]*_align_shift. Only works if *[x/y]*_align is specified. This “nudges” the item from its base positioning, and works appropriately for both negative and positive values, if that does not otherwise violate boundaries. This is best used for things like pushing something off the hard edge of the document’s margins.

Embedding Images

Putting images onto your certificate is just done via a stampable with an `img` key. An example from above:

```
{
    "height": null,
    "img": "https://www.google.com/images/srpr/logo11w.png",
```

```
    "stamp_type": "image",
    "width": 40,
    "x": null,
    "x_align": "center",
    "x_align_shift": null,
    "x_percent": null,
    "y": null,
    "y_align": null,
    "y_align_shift": null,
    "y_percent": 5
},
```

Placement rules are the same as for text as described in stampable positioning. The image must reside at a URL accessible from the API server.

Whenever possible, use PNGs for your embedded images. JPEGs do work in some cases, but are more unreliable for the PDF type.

The `img` tag can take dynamic variables, like in the following:

```
{
    "height": null,
    "img": "https://www.google.com/images/srpr/logo{{SCORE}}.png",
    "stamp_type": "image",
    "width": 40,
    "x": null,
    "x_align": "center",
    "x_align_shift": null,
    "x_percent": null,
    "y": null,
    "y_align": null,
    "y_align_shift": null,
    "y_percent": 5
},
```

Send a `?SCORE=...` as part of the URL and the img will be resolved with that variable before attempting to fetch.

CUSTOMIZING EXISTING PAGES

Any of the pages in Enfix can be completely customized to your needs.

Template Language

All templates in Enfix are built with Jinja2: http://jinja.pocoo.org/docs/2.10/

General Structure

Many of the templates themselves inherit from other templates. The application structure is such that you may choose at any point whether to inherit from other templates within your own space, or the templates from the base application.

All base application templates can be found in `templates/_default`.

Overriding the Base Template

`base.html` represents the general layout that all pieces of the site are based on. It's completely possible to change this file in your own implementations, and all other files you have not overridden will still use this extension.

Common Blocks

- `{% block js_global %}` For global overrides that you want to apply irrespective of per-page customization needs.
- `{% block js_page %}`. For per-page JS behavior. Most page templates will put their custom behaviors here, and can be replaced by overriding this block.
- `{% block css_global %}`. For global CSS overrides that you want to apply irrespective of per-page customization needs. Custom static CSS files are still considered preferable.
- `{% block css_page %}`. For per-page CSS behavior. Unlike `js_page`, there's rarely inlined CSS to be inherited from templates, and is here only for the customizer's convenience.
- `{% block content %}`. The general central area of the page, inside all major containers and menus. Most pages will contain most of their functional behavior in this block.

A Simple Override Example

Let's add a small amount of Javascript to run on every page of the site, like a Google Analytics tracker.

1. Begin by creating a directory under templates named as your site's domain. For this example, we'll use `example.com`.
2. Create a file with the full path of `templates/example.com/base.html`

Populate it with the following code:

```
{% extends "_default/base.html" %}
{% block js_global %}
<script type="text/javascript">
console.log("Hello world!");
</script>
{% endblock %}
```

3.

Now load any page on your site. In your console, you should see the `Hello world!` execution output.

ADDING NEW PAGES

If you have custom materials that don't make sense to add to an existing page, you can inject a new routed page into the application structure. New pages can either be behind a login (private pages) or visible without a login (public pages).

Public Pages

1. To install a public page, in your `templates/<your_host>/` folder, make a folder named `custom`.
2. In that `custom` folder, make a folder named `public`
3. Create your public file. The required extension on the file is .html. Nested directories are not supported.
4. Open your browser to https://yoursite.com/c/`file` . Your file should be visible and render appropriately.
5. These are rendered templates through the Jinja templating system, so be careful about braces.
6. Available variables:
    1. api_url: The fully qualified URL for API requests.
    2. app_token: The application token for API requests.

Private Pages

1. To install a private page, in your `templates/<your_host>/` folder, make a folder named `custom`.
2. In that `custom` folder, make a folder named `private`
3. Create your private file. The required extension on the file is .html. Nested directories are not supported.
4. Open your browser to https://yoursite.com/t/`file` . Your file should be visible and render appropriately if you are logged in. Not logged in users will be redirected to the login page.
5. These are rendered templates through the Jinja templating system, so be careful about braces.
6. Available variables:
    - api_url: The fully qualified URL for API requests.
    - app_token: The application token for API requests.
    - token: The user token for API requests.

API ENDPOINTS

Activity Sessions

Note: Every endpoint listed in this document is managing JSON objects, so the path structure shown for nested keys is represented in a JSON-style access pattern.

Root Path: `/api/v1/activity_esssions/`

Model

Key	Type	Default	Description
course	string	<colander.required>	
user	string	<colander.required>	
score	number		
completed	bool	false	
passed	bool	false	
attempt_open_until	datetime		
manual_audit_date	datetime		
custom	json		
session	string	<unused>	
sub_identifier	string	<unused>	
session_closed	bool	<unused>	

List

Return a list of this collection's items, matching the filter parameters if provided.

HTTP Verb: `GET`

Endpoint: `/api/v1/activity_sessions/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/activity_sessions/
```

Example Response:


```
[
{
  "_id": "54ac1e5140df567c3297d4b8",
  "date_added": "2015-01-06 12:41:37",
  "manual_audit_date": null,
  "date_updated": "2015-01-06 12:41:37",
  "sub_identifier": "",
  "attempt_open_until": null,
  "passed": true,
  "completed": false,
  "session_closed": false,
  "course": "53473891a64c9e7c66b2390b",
  "session": "",
  "score": 101,
  "custom": {},
  "user": "51b1e833bd7a844089b652aa"
},
{
  "_id": "56b012547bb6b18a0dad6ddd",
  "score": 82,
  "attempt_open_until": null,
  "date_updated": "2016-02-01 21:20:27",
  "sub_identifier": "51b1e833bd7a844089b652aa_testerf",
  "session_closed": false,
  "custom": {
    "scorm_data": "{\"cmi.core.score.max\": \"\",
\"cmi.comments_from_lms\": \"No comment\", \"cmi.comments\": \"\",
\"cmi.core.score.raw\": \"82\", \"cmi.core.student_id\":
\"51b1e833bd7a844089b652aa\", \"cmi.student_preference.language\":
\"\", \"cmi.student_preference.text\": \"\",
\"cmi.core.lesson_location\": \"\", \"cmi.student_preference.speed\":
\"\", \"cmi.launch_data\": \"\", \"cmi.student_preference.audio\":
\"\", \"cmi.core.score.min\": \"\", \"cmi.core.total_time\":
\"0000:00:00\", \"cmi.student_data.mastery_score\": \"\",
\"cmi.core.credit\": \"credit\", \"cmi.core.entry\": \"\",
\"cmi.objectives._count\": \"0\",
\"cmi.student_data.max_time_allowed\": \"\", \"cmi.core.student_name\":
\"gav163\", \"cmi.interactions._count\": \"0\",
\"cmi.core.lesson_status\": \"passed\",
\"cmi.student_data.time_limit_action\": \"\", \"cmi.core.lesson_mode\":
\"normal\", \"cmi.suspend_data\": \"\"}"
  },
```

```
  "course": "51d5d55dbd7a84058e7bf61c",
  "session": "",
  "user": "51b1e833bd7a844089b652aa",
  "passed": true,
  "manual_audit_date": null,
  "date_added": "2016-02-01 21:20:04",
  "completed": false
}
]
```


Get One

Return the specified collection item.

HTTP Verb: `GET`

Endpoint: `/api/v1/activity_sessions/[id]`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/activity_sessions/[id]
```

Example Response:


```
{
  "_id": "56b012547bb6b18a0dad6ddd",
  "score": 82,
  "attempt_open_until": null,
  "date_updated": "2016-02-01 21:20:27",
  "sub_identifier": "51b1e833bd7a844089b652aa_testerf",
  "session_closed": false,
  "custom": {
    "scorm_data": "{\"cmi.core.score.max\": \"\",
\"cmi.comments_from_lms\": \"No comment\", \"cmi.comments\": \"\",
\"cmi.core.score.raw\": \"82\", \"cmi.core.student_id\":
\"51b1e833bd7a844089b652aa\", \"cmi.student_preference.language\":
\"\", \"cmi.student_preference.text\": \"\",
\"cmi.core.lesson_location\": \"\", \"cmi.student_preference.speed\":
\"\", \"cmi.launch_data\": \"\", \"cmi.student_preference.audio\":
\"\", \"cmi.core.score.min\": \"\", \"cmi.core.total_time\":
\"0000:00:00\", \"cmi.student_data.mastery_score\": \"\",
\"cmi.core.credit\": \"credit\", \"cmi.core.entry\": \"\",
\"cmi.objectives._count\": \"0\",
\"cmi.student_data.max_time_allowed\": \"\", \"cmi.core.student_name\":
\"gav163\", \"cmi.interactions._count\": \"0\",
\"cmi.core.lesson_status\": \"passed\",
```

```
\"cmi.student_data.time_limit_action\": \"\", \"cmi.core.lesson_mode\":
\"normal\", \"cmi.suspend_data\": \"\"}"
  },
  "course": "51d5d55dbd7a84058e7bf61c",
  "session": "",
  "user": "51b1e833bd7a844089b652aa",
  "passed": true,
  "manual_audit_date": null,
  "date_added": "2016-02-01 21:20:04",
  "completed": false
}
```

Create

Store a new instance of the object in the system. All required fields must be specified.

HTTP Verb: `POST`

Endpoint: `/api/v1/activity_sessions/`

Example Request (cURL):

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     -d '{"course": "51d5d55dbd7a84058e7bf61c", "user":
"51b1e833bd7a844089b652aa"}'
     https://<yournamehere>.enfixlp.com/api/v1/activity_sessions/
```

Example Response:

```
{
    "resource_url":
"https://<yournamehere>.enfixlp.com/api/v1/activity_sessions/51b1e78dbd
7a844089b652a9/",
    "_id": "51b1e78dbd7a844089b652a9"
}
```

Replace

This is a full model replacement of the specified item, every required field must be specified.

HTTP Verb: `PUT`

Endpoint: `/api/v1/activity_sessions/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     -d '{"course": "51d5d55dbd7a84058e7bf61c", "user":
"51b1e833bd7a844089b652aa"}'
     https://<yournamehere>.enfixlp.com/api/v1/activity_sessions/[id]
```

Example Response:

```
No body, Status Code 204.
```

Update

On an update, you may specify only the fields you wish to update. All other fields will be preserved as currently stored.

HTTP Verb: `PATCH`

Endpoint: `/api/v1/activity_sessions/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     -d '{"passed": false}'
     https://<yournamehere>.enfixlp.com/api/v1/activity_sessions/[id]
```

Example Response:

```
No body, Status Code 204.
```

Delete

Permanently remove an item from the collection.

HTTP Verb: `DELETE`

Endpoint: `/api/v1/activity_sessions/[id]`

Example Request (cURL):

```
curl -X DELETE -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/activity_sessions/[id]
```

Example Response:

```
No body, Status Code 200
```

Course Categories

Note: Every endpoint listed in this document is managing JSON objects, so the path structure shown for nested keys is represented in a JSON-style access pattern.

Root Path: `/course_categories/`

Model

Key	Type	Default	Description
name	string	<colander.required>	
category_type	string		
description	string		
parent	string		
courses	[]string	[]	

List

Return a list of this collection's items, matching the filter parameters if provided.

HTTP Verb: `GET`

Endpoint: `/api/v1/course_categories/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
```

```
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/course_categories/
```

Example Response:

```
[
{
    "courses": [],
    "category_type": "",
    "name": "High School Mathematics",
    "parent": "",
    "description": ""
}
]
```

Get One

Return the specified collection item.

HTTP Verb: `GET`

Endpoint: `/api/v1/course_categories/[id]`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/course_categories/[id]
```

Example Response:

```
{
    "courses": [],
    "category_type": "",
```

```
    "name": "High School Mathematics",

    "parent": "",

    "description": ""

}
```

Create

Store a new instance of the object in the system. All required fields must be specified.

HTTP Verb: `POST`

Endpoint: `/api/v1/course_categories/`

Example Request (cURL):

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"

     -d '{"courses": [], "category_type": "", "name": "High School
Mathematics", "parent": "", "description": ""}'

     https://<yournamehere>.enfixlp.com/api/v1/course_categories/
```

Example Response:

```
{

    "resource_url":
"https://<yournamehere>.enfixlp.com/api/v1/course_categories/51b1e78dbd
7a844089b652a9/",

    "_id": "51b1e78dbd7a844089b652a9"

}
```

Replace

This is a full model replacement of the specified item, every required field must be specified.

HTTP Verb: `PUT`

Endpoint: `/api/v1/course_categories/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"

     -d '{"courses": [], "category_type": "", "name": "High School
Mathematics", "parent": "", "description": ""}'

     https://<yournamehere>.enfixlp.com/api/v1/course_categories/[id]
```

Example Response:

```
No body, Status Code 204.
```

Update

On an update, you may specify only the fields you wish to update. All other fields will be preserved as currently stored.

HTTP Verb: `PATCH`

Endpoint: `/api/v1/course_categories/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"

     -d '{"courses": []}'

     https://<yournamehere>.enfixlp.com/api/v1/course_categories/[id]
```

Example Response:

```
No body, Status Code 204.
```

Delete

Permanently remove an item from the collection.

HTTP Verb: `DELETE`

Endpoint: `/api/v1/course_categories/[id]`

Example Request (cURL):

```
curl -X DELETE -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/course_categories/[id]
```

Example Response:

```
No body, Status Code 200.
```

Courses

Note: Every endpoint listed in this document is managing JSON objects, so the path structure shown for nested keys is represented in a JSON-style access pattern.

Root Path: `/api/v1/courses/`

Model

Key	Type	Default	Description
title	string	<colander.required>	
course_type	string		
description	string		
active	bool	false	

requires_enrollment	bool	true	
enrolled_groups	[]string	[]	
enrolled_users	[]string	[]	
manual_ordering_index	int	0	
due_type	string		
due_exact_date	datetime		
due_days_from_enrollment	int	0	
display_options.show_navigation_panel	bool	true	
display_options.show_navigation_header	bool	true	
display_options.navigation_header_height	int	40	
display_options.navigation_panel_width	int	200	

display_options.player_launch_method	int	1	
display_options.sco_launch_method	int	1	
display_options.sco_window_size_type	int	1	
display_options.sco_window_width	int	800	
display_options.sco_window_height	int	600	
display_options.player_window_size_type	int	1	
display_options.player_window_width	int	1000	
display_options.player_window_height	int	640	
behavior.allow_score_to_decrease	bool	true	
behavior.allow_score_change_after_status_passed	bool	true	
behavior.allow_score_change_after_status_failed	bool	true	

behavior.allow_score_change_after_status_completed	bool	true	
behavior.allow_status_change_after_status_passed	bool	true	
behavior.allow_status_change_after_status_failed	bool	true	
behavior.allow_status_change_after_status_completed	bool	true	

List

Return a list of this collection's items, matching the filter parameters if provided.

HTTP Verb: `GET`

Endpoint: `/api/v1/courses/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/courses/
```

Example Response:

```
[
{
    "requires_enrollment": true,
    "display_options": {
        "player_launch_method": 1,
        "sco_window_width": 800,
```

```
        "sco_window_height": 600,
        "navigation_header_height": 40,
        "player_window_width": 1000,
        "show_navigation_panel": true,
        "navigation_panel_width": 200,
        "show_navigation_header": true,
        "sco_launch_method": 1,
        "player_window_height": 640,
        "player_window_size_type": 1,
        "sco_window_size_type": 1
    },
    "enrolled_groups": [],
    "title": "Business Management Skills",
    "manual_ordering_index": 0,
    "course_type": "",
    "due_exact_date": "",
    "enrolled_users": [],
    "due_type": "",
    "behavior": {
        "allow_status_change_after_status_completed": true,
        "allow_score_change_after_status_failed": true,
        "allow_score_change_after_status_completed": true,
        "allow_status_change_after_status_failed": true,
        "allow_score_change_after_status_passed": true,
        "allow_status_change_after_status_passed": true,
        "allow_score_to_decrease": true
    },
```

```
    "active": false,
    "due_days_from_enrollment": 0,
    "metadata": {},
    "description": ""
}
]
```

Get One

Return the specified collection item.

HTTP Verb: `GET`

Endpoint: `/api/v1/courses/[id]`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/courses/[id]
```

Example Response:

```
{
    "requires_enrollment": true,
    "display_options": {
        "player_launch_method": 1,
        "sco_window_width": 800,
        "sco_window_height": 600,
        "navigation_header_height": 40,
        "player_window_width": 1000,
        "show_navigation_panel": true,
        "navigation_panel_width": 200,
```

```
        "show_navigation_header": true,
        "sco_launch_method": 1,
        "player_window_height": 640,
        "player_window_size_type": 1,
        "sco_window_size_type": 1
    },
    "enrolled_groups": [],
    "title": "Business Management Skills",
    "manual_ordering_index": 0,
    "course_type": "",
    "due_exact_date": "",
    "enrolled_users": [],
    "due_type": "",
    "behavior": {
        "allow_status_change_after_status_completed": true,
        "allow_score_change_after_status_failed": true,
        "allow_score_change_after_status_completed": true,
        "allow_status_change_after_status_failed": true,
        "allow_score_change_after_status_passed": true,
        "allow_status_change_after_status_passed": true,
        "allow_score_to_decrease": true
    },
    "active": false,
    "due_days_from_enrollment": 0,
    "metadata": {},
    "description": ""
}
```

Create

Store a new instance of the object in the system. All required fields must be specified.

HTTP Verb: `POST`

Endpoint: `/api/v1/courses/`

Example Request (cURL):

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"

     -d '{"requires_enrollment": true, "display_options":
{"player_launch_method": 1, "sco_window_width": 800,
"sco_window_height": 600, "navigation_header_height": 40,
"player_window_width": 1000, "show_navigation_panel": true,
"navigation_panel_width": 200, "show_navigation_header": true,
"sco_launch_method": 1, "player_window_height": 640,
"player_window_size_type": 1, "sco_window_size_type": 1},
"enrolled_groups": [], "title": "Business Management Skills",
"manual_ordering_index": 0, "course_type": "", "due_exact_date": "",
"enrolled_users": [], "due_type": "", "behavior":
{"allow_status_change_after_status_completed": true,
"allow_score_change_after_status_failed": true,
"allow_score_change_after_status_completed": true,
"allow_status_change_after_status_failed": true,
"allow_score_change_after_status_passed": true,
"allow_status_change_after_status_passed": true,
"allow_score_to_decrease": true}, "active": false,
"due_days_from_enrollment": 0, "metadata": {}, "description": ""}'

     https://<yournamehere>.enfixlp.com/api/v1/courses/
```

Example Response:

```
{

    "resource_url":
"https://<yournamehere>.enfixlp.com/api/v1/courses/51b1e78dbd7a844089b6
52a9/",

    "_id": "51b1e78dbd7a844089b652a9"

}
```

Replace

This is a full model replacement of the specified item, every required field must be specified.

HTTP Verb: `PUT`

Endpoint: `/api/v1/courses/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"

     -d '{"requires_enrollment": true, "display_options":
{"player_launch_method": 1, "sco_window_width": 800,
"sco_window_height": 600, "navigation_header_height": 40,
"player_window_width": 1000, "show_navigation_panel": true,
"navigation_panel_width": 200, "show_navigation_header": true,
"sco_launch_method": 1, "player_window_height": 640,
"player_window_size_type": 1, "sco_window_size_type": 1},
"enrolled_groups": [], "title": "Business Management Skills",
"manual_ordering_index": 0, "course_type": "", "due_exact_date": "",
"enrolled_users": [], "due_type": "", "behavior":
{"allow_status_change_after_status_completed": true,
"allow_score_change_after_status_failed": true,
"allow_score_change_after_status_completed": true,
"allow_status_change_after_status_failed": true,
"allow_score_change_after_status_passed": true,
"allow_status_change_after_status_passed": true,
"allow_score_to_decrease": true}, "active": false,
"due_days_from_enrollment": 0, "metadata": {}, "description": ""}'

     https://<yournamehere>.enfixlp.com/api/v1/courses/[id]
```

Example Response:

```
No body, Status Code 204.
```

Update

On an update, you may specify only the fields you wish to update. All other fields will be preserved as currently stored.

HTTP Verb: `PATCH`

Endpoint: `/api/v1/courses/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     -d '{"requires_enrollment": true}'
     https://<yournamehere>.enfixlp.com/api/v1/courses/[id]
```

Example Response:

```
No body, Status Code 204.
```

Delete

Permanently remove an item from the collection.

HTTP Verb: `DELETE`

Endpoint: `/api/v1/courses/[id]`

Example Request (cURL):

```
curl -X DELETE -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/courses/[id]
```

Example Response:

```
No body, Status Code 200.
```

Upload

Upload a course package into a given course container.

HTTP Verb: `POST`

Endpoint: `/api/v1/courses/[id]/upload/`

Example Request (cURL):

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     --data-binary @pif.zip
     https://<yournamehere>.enfixlp.com/api/v1/courses/[id]/upload/
```

Example Response:

```
No body, Status Code 204.
```

Enrollment Endpoints

Note: Every endpoint listed in this document is managing JSON objects, so the path structure shown for nested keys is represented in a JSON-style access pattern.

User - Courses Enrolled In

Return a list of all the courses a user is currently enrolled in.

HTTP Verb: `GET`

Endpoint: `/api/v1/views/user/<string:user_id>/courses_enrolled_in/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"

https://<yournamehere>.enfixlp.com/api/v1/views/user/<string:user_id>/courses_enrolled_in/
```

Example Response:

```
[
{
    "_id": "5330f129a64c9e20eb2732a5",
    "requires_enrollment": false,
```

```
    "display_options": {
        "player_launch_method": 1,
        "navigation_panel_width": 200,
        "sco_window_width": 800,
        "show_navigation_header": true,
        "sco_window_height": 600,
        "navigation_header_height": 40,
        "show_navigation_panel": true,
        "sco_launch_method": 2,
        "sco_window_size_type": 1
    },
    "enrolled_groups": [],
    "title": "Text SCO 2",
    "date_updated": "2014-06-29 19:46:51",
    "manual_ordering_index": 0,
    "course_type": "scorm",
    "enrolled_users": [],
    "active": false,
    "date_added": "2014-03-24 22:59:53",
    "metadata": {},
    "description": ""
},
{
    "_id": "5bf326aa2fd84e7981bbbea3",
    "requires_enrollment": false,
    "display_options": {
        "player_launch_method": 1,
```

```
        "navigation_panel_width": 200,
        "sco_window_width": 800,
        "show_navigation_header": true,
        "sco_window_height": 600,
        "navigation_header_height": 40,
        "player_window_width": 1000,
        "show_navigation_panel": true,
        "sco_launch_method": 1,
        "player_window_height": 640,
        "player_window_size_type": 1,
        "sco_window_size_type": 1
    },
    "enrolled_groups": [
        "5a6605c87bb6b15e342c01d1"
    ],
    "title": "Multi SCO 1",
    "date_updated": "2018-11-19 16:13:01",
    "include_subgroups": [],
    "manual_ordering_index": 0,
    "course_type": "scorm",
    "due_exact_date": "",
    "enrolled_users": [],
    "due_type": "",
    "behavior": {
        "allow_status_change_after_status_completed": true,
        "allow_score_change_after_status_failed": true,
        "allow_score_change_after_status_completed": true,
```

```
        "allow_status_change_after_status_failed": true,
        "allow_score_change_after_status_passed": true,
        "scorm_sco_rollup_mode": "last",
        "allow_status_change_after_status_passed": true,
        "allow_score_to_decrease": true
    },
    "active": true,
    "date_added": "2018-11-19 16:10:02",
    "due_days_from_enrollment": 0,
    "metadata": {},
    "description": "."
}
]
```


User - Courses Enrolled In With Status

Return a list of all the courses a user is currently enrolled in, and any high-level status/score information in the “rollup_status” block.

HTTP Verb: `GET`

Endpoint: `/api/v1/views/user/<string:user_id>/courses_enrolled_in/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"

https://<yournamehere>.enfixlp.com/api/v1/views/user/<string:user_id>/courses_enrolled_in/
```

Example Response:


```
[
```

```
{
    "_id": "51d5d55dbd7a84058e7bf61c",
    "requires_enrollment": false,
    "display_options": {
        "player_launch_method": 1,
        "navigation_panel_width": 221,
        "sco_window_width": 550,
        "show_navigation_header": false,
        "sco_window_height": 551,
        "navigation_header_height": 51,
        "player_window_width": 400,
        "show_navigation_panel": false,
        "sco_launch_method": 1,
        "player_window_height": 300,
        "player_window_size_type": 1,
        "sco_window_size_type": 1
    },
    "enrolled_groups": [
        "5a6605c87bb6b15e342c01af",
        "5a6605c87bb6b15e342c01b1",
        "5a6605c87bb6b15e342c01b3"
    ],
    "title": "SCO Tester",
    "date_updated": "2018-01-29 07:47:15",
    "manual_ordering_index": 200,
    "course_type": "scorm",
    "due_exact_date": "2018-01-31 00:00:00",
```

```
    "enrolled_users": [],
    "due_type": "exact_date",
    "behavior": {
        "allow_status_change_after_status_completed": true,
        "allow_score_change_after_status_failed": true,
        "allow_score_change_after_status_completed": true,
        "allow_status_change_after_status_failed": true,
        "allow_score_change_after_status_passed": true,
        "allow_status_change_after_status_passed": true,
        "allow_score_to_decrease": true
    },
    "active": true,
    "date_added": "2014-03-14T19:50:48.517000",
    "due_days_from_enrollment": 360,
    "metadata": {},
    "description": "Description required here.",
    "calculated_due_date": "2018-01-31 00:00:00",
    "rollup_status": {
        "status": "C",
        "score": 82.0,
        "date_added": "2018-07-31 23:37:04",
        "date_updated": "2019-11-23 08:01:46",
        "total_wall_time": 578865
    }
},
{
    "_id": "5145eb3abd7a844438bb931c",
```

```
    "requires_enrollment": false,
    "display_options": {
        "player_launch_method": 2,
        "navigation_panel_width": 200,
        "sco_window_width": 800,
        "show_navigation_header": false,
        "sco_window_height": 600,
        "navigation_header_height": 40,
        "show_navigation_panel": true,
        "sco_launch_method": 1,
        "sco_window_size_type": 1
    },
    "enrolled_groups": [
        "51413a49bd7a841b9ca9528b",
        "51413a67bd7a841b9ca9528c",
        "51101a73bd7a841143c4cb9c",
        "51101c07bd7a8412229d1f31"
    ],
    "title": "Diagnostic SCO",
    "date_updated": "2014-07-17 11:31:09",
    "manual_ordering_index": 100,
    "course_type": "scorm",
    "enrolled_users": [],
    "active": true,
    "date_added": "2014-03-14 19:50:48",
    "metadata": {},
    "description": "",
```

```
    "rollup_status": {
        "status": "N",
        "score": "",
        "date_added": "",
        "date_updated": "",
        "total_wall_time": -1
    }
},
{
    "_id": "5dea70cfe6c8349439869e42",
    "title": "AICC Test 1",
    "course_type": "aicc",
    "description": ".",
    "active": false,
    "requires_enrollment": false,
    "enrolled_groups": [],
    "enrolled_users": [],
    "include_subgroups": [],
    "manual_ordering_index": 0,
    "due_type": "",
    "due_exact_date": "",
    "due_days_from_enrollment": 0,
    "aicc_launch_url": "",
    "metadata": {},
    "display_options": {
        "show_navigation_panel": true,
        "show_navigation_header": true,
```

```
        "navigation_header_height": 40,
        "navigation_panel_width": 200,
        "player_launch_method": 1,
        "sco_launch_method": 1,
        "sco_window_size_type": 1,
        "sco_window_width": 800,
        "sco_window_height": 600,
        "player_window_size_type": 1,
        "player_window_width": 1000,
        "player_window_height": 640
    },
    "behavior": {
        "allow_score_to_decrease": true,
        "allow_score_change_after_status_passed": true,
        "allow_score_change_after_status_failed": true,
        "allow_score_change_after_status_completed": true,
        "allow_status_change_after_status_passed": true,
        "allow_status_change_after_status_failed": true,
        "allow_status_change_after_status_completed": true,
        "scorm_sco_rollup_mode": ""
    },
    "date_added": "2019-12-06 10:16:31",
    "date_updated": "2019-12-06 10:16:44",
    "rollup_status": {
        "status": "N",
        "score": "",
        "date_added": "",
```

```
        "date_updated": "",
        "total_wall_time": -1
    }
}
]
```

Course - Users Enrolled In

Return a list of all the users enrolled in a course.

HTTP Verb: `GET`

Endpoint: `/api/v1/views/course/<string:course_id>/users_enrolled_in/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
    -H "Content-Type: application/json"

https://<yournamehere>.enfixlp.com/api/v1/views/course/<string:course_id>/users_enrolled_in/
```

Example Response:

```
[
{
    "_id": "51293841bd7a8494f0a3369f",
    "username": "test12",
    "name": "Test User 1",
    "roles": [
        "557d927a40df56c7fe5a89f9"
    ],
    "date_updated": "2018-10-11 11:07:16",
    "date_added": "2013-03-17 23:52:26",
```

```
    "active": true,
    "_password": "...",
    "email": "test14@test14.com",
},
{
    "_id": "51299265bd7a84af48bd2a51",
    "_password": "...",
    "date_added": "2013-03-17 23:52:26",
    "date_updated": "2013-02-23 23:09:09",
    "email": "test13@test13.com",
    "groups": [],
    "name": "Test User 13",
    "roles": [],
    "username": "test13"
}
]
```


Groups

Note: Every endpoint listed in this document is managing JSON objects, so the path structure shown for nested keys is represented in a JSON-style access pattern.

Root Path: `/groups/`

Model

Key	Type	Default	Description
name	string	<colander.required>	

group_type	string		
description	string		
parent	string		
members	[]string	[]	
managers	[]string	[]	

List

Return a list of this collection's items, matching the filter parameters if provided.

HTTP Verb: `GET`

Endpoint: `/api/v1/groups/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/groups/
```

Example Response:

```
[
{
    "managers": [],
    "name": "Heavy Equipment Drivers",
    "parent": "",
```

```
    "members": [],
    "group_type": "",
    "description": ""
}
]
```

Get One

Return the specified collection item.

HTTP Verb: `GET`

Endpoint: `/api/v1/groups/[id]`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/groups/[id]
```

Example Response:

```
{
    "managers": [],
    "name": "Heavy Equipment Drivers",
    "parent": "",
    "members": [],
    "group_type": "",
    "description": ""
}
```

Create

Store a new instance of the object in the system. All required fields must be specified.

HTTP Verb: `POST`

Endpoint: `/api/v1/groups/`

Example Request (cURL):

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"

     -d '{"managers": [], "name": "Heavy Equipment Drivers", "parent":
"", "members": [], "group_type": "", "description": ""}'

     https://<yournamehere>.enfixlp.com/api/v1/groups/
```

Example Response:

```
{

    "resource_url":
"https://<yournamehere>.enfixlp.com/api/v1/groups/51b1e78dbd7a844089b65
2a9/",

    "_id": "51b1e78dbd7a844089b652a9"

}
```

Replace

This is a full model replacement of the specified item, every required field must be specified.

HTTP Verb: `PUT`

Endpoint: `/api/v1/groups/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"

     -d '{"managers": [], "name": "Heavy Equipment Drivers", "parent":
"", "members": [], "group_type": "", "description": ""}'

     https://<yournamehere>.enfixlp.com/api/v1/groups/[id]
```

Example Response:

```
No body, Status Code 204.
```

Update

On an update, you may specify only the fields you wish to update. All other fields will be preserved as currently stored.

HTTP Verb: `PATCH`

Endpoint: `/api/v1/groups/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     -d '{"managers": []}'
     https://<yournamehere>.enfixlp.com/api/v1/groups/[id]
```

Example Response:

```
No body, Status Code 204.
```

Delete

Permanently remove an item from the collection.

HTTP Verb: `DELETE`

Endpoint: `/api/v1/groups/[id]`

Example Request (cURL):

```
curl -X DELETE -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/groups/[id]
```

Example Response:

```
No body, Status Code 200.
```

Reporting Endpoints

Note: Every endpoint listed in this document is managing JSON objects, so the path structure shown for nested keys is represented in a JSON-style access pattern.

Scores Per Course

Return a list of all the learners who are enrolled in the course, and specifies their completions and scores for the course.

HTTP Verb: `GET`

Endpoint: `/api/v1/reports/scores_per_course/<string:course_id>/?show_users=1`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"


https://<yournamehere>.enfixlp.com/api/v1/reports/scores_per_course/<st
ring:course_id>/?show_users=1
```

Example Response:

```
[

{

    "user": {

        "_id": "51293841bd7a8494f0a3369f",

        "username": "tmp12",

        "name": "Test User 1",

        "email": "tmp12@tmp12.com",

    },
```

```
    "score": "",
    "score_date": "",
    "total_wall_time": -1,
    "passed": false
},
{
    "user": {
        "_id": "51b1e833bd7a844089b652aa",
        "username": "tmp13",
        "name": "Test User 2",
        "email": "a@e.com",
    },
    "score": 82,
    "total_wall_time": -1,
    "score_date": "2016-02-01 21:20:27",
    "passed": true
}
]
```


Student Transcript

Return a list of all the courses a learner is enrolled in.

HTTP Verb: `GET`

Endpoint: `/api/v1/reports/student_transcript/<string:user_id>/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
```

```
https://<yournamehere>.enfixlp.com/api/v1/reports/student_transcript/<string:user_id>/
```

Example Response:


```
[
{
    "course": {
        "_id": "51d5d55dbd7a84058e7bf61c",
        "title": "SCO Tester"
    },
    "started": true,
    "passed": true,
    "score": 82.0,
    "date_achieved": "2019-11-23 08:01:46"
},
{
    "course": {
        "_id": "5145eb3abd7a844438bb931c",
        "title": "Diagnostic SCO"
    },
    "started": false,
    "passed": false,
    "score": "",
    "date_achieved": ""
}
]
```

Roles

Note: Every endpoint listed in this document is managing JSON objects, so the path structure shown for nested keys is represented in a JSON-style access pattern.

Root Path: `/roles/`

Model

Key	Type	Default	Description
role	string	<colander.required>	
description	string		
allow_all	bool	false	
permissions	[]string	[]	

List

Return a list of this collection's items, matching the filter parameters if provided.

HTTP Verb: `GET`

Endpoint: `/api/v1/roles/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/roles/
```

Example Response:

```
[
{
    "allow_all": false,
    "role": "Plant Manager",
    "description": "",
    "permissions": []
}
]
```

Get One

Return the specified collection item.

HTTP Verb: `GET`

Endpoint: `/api/v1/roles/[id]`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/roles/[id]
```

Example Response:

```
{
    "allow_all": false,
    "role": "Plant Manager",
    "description": "",
    "permissions": []
}
```

Create

Store a new instance of the object in the system. All required fields must be specified.

HTTP Verb: `POST`

Endpoint: `/api/v1/roles/`

Example Request (cURL):

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"

     -d '{"allow_all": false, "role": "Plant Manager", "description":
"", "permissions": []}'

     https://<yournamehere>.enfixlp.com/api/v1/roles/
```

Example Response:

```
{

    "resource_url":
"https://<yournamehere>.enfixlp.com/api/v1/roles/51b1e78dbd7a844089b652
a9/",

    "_id": "51b1e78dbd7a844089b652a9"

}
```

Replace

This is a full model replacement of the specified item, every required field must be specified.

HTTP Verb: `PUT`

Endpoint: `/api/v1/roles/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"

     -d '{"allow_all": false, "role": "Plant Manager", "description":
"", "permissions": []}'
```

```
    https://<yournamehere>.enfixlp.com/api/v1/roles/[id]
```

Example Response:

```
No body, Status Code 204.
```

Update

On an update, you may specify only the fields you wish to update. All other fields will be preserved as currently stored.

HTTP Verb: `PATCH`

Endpoint: `/api/v1/roles/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
    -H "Content-Type: application/json"
    -d '{"allow_all": false}'
    https://<yournamehere>.enfixlp.com/api/v1/roles/[id]
```

Example Response:

```
No body, Status Code 204.
```

Delete

Permanently remove an item from the collection.

HTTP Verb: `DELETE`

Endpoint: `/api/v1/roles/[id]`

Example Request (cURL):

```
curl -X DELETE -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
    -H "Content-Type: application/json"
```

```
https://<yournamehere>.enfixlp.com/api/v1/roles/[id]
```

Example Response:

```
No body, Status Code 200.
```

Users

Note: Every endpoint listed in this document is managing JSON objects, so the path structure shown for nested keys is represented in a JSON-style access pattern.

Root Path: `/users/`

Model

Key	Type	Default	Description
username	string	<colander.required>	
password	string		
name	string	<colander.required>	
email	string	<colander.required>	
active	bool	true	
roles	[]string	[]	

List

Return a list of this collection's items, matching the filter parameters if provided.

HTTP Verb: `GET`

Endpoint: `/api/v1/users/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/users/
```

Example Response:

```
[
{
    "username": "johndoe",
    "name": "John Doe",
    "roles": [],
    "active": true,
    "password": "",
    "email": "johndoe@example.com",
    "metadata": {}
}
]
```

Get One

Return the specified collection item.

HTTP Verb: `GET`

Endpoint: `/api/v1/users/[id]`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/users/[id]
```

Example Response:

```
{
    "username": "johndoe",
    "name": "John Doe",
    "roles": [],
    "active": true,
    "password": "",
    "email": "johndoe@example.com",
    "metadata": {}
}
```

Create

Store a new instance of the object in the system. All required fields must be specified.

HTTP Verb: `POST`

Endpoint: `/api/v1/users/`

Example Request (cURL):

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     -d '{"username": "johndoe", "name": "John Doe", "roles": [],
"active": true, "password": "", "email": "johndoe@example.com",
"metadata": {}}'
     https://<yournamehere>.enfixlp.com/api/v1/users/
```

Example Response:

```
{

    "resource_url":
"https://<yournamehere>.enfixlp.com/api/v1/users/51b1e78dbd7a844089b652
a9/",

    "_id": "51b1e78dbd7a844089b652a9"

}
```

Replace

This is a full model replacement of the specified item, every required field must be specified.

HTTP Verb: `PUT`

Endpoint: `/api/v1/users/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"

     -H "Content-Type: application/json"

     -d '{"username": "johndoe", "name": "John Doe", "roles": [],
"active": true, "password": "", "email": "johndoe@example.com",
"metadata": {}}'

     https://<yournamehere>.enfixlp.com/api/v1/users/[id]
```

Example Response:

```
No body, Status Code 204.
```

Update

On an update, you may specify only the fields you wish to update. All other fields will be preserved as currently stored.

HTTP Verb: `PATCH`

Endpoint: `/api/v1/users/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     -d '{"username": "johndoe"}'
     https://<yournamehere>.enfixlp.com/api/v1/users/[id]
```

Example Response:

```
No body, Status Code 204.
```

Delete

Permanently remove an item from the collection.

HTTP Verb: `DELETE`

Endpoint: `/api/v1/users/[id]`

Example Request (cURL):

```
curl -X DELETE -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     https://<yournamehere>.enfixlp.com/api/v1/users/[id]
```

Example Response:

```
No body, Status Code 200.
```

Webhooks

Note: Every endpoint listed in this document is managing JSON objects, so the path structure shown for nested keys is represented in a JSON-style access pattern.

Root Path: `/webhooks/`

Model

Key	Type	Default	Description
endpoint	string	<colander .required>	The name of an Enfix endpoint. ex: “courses”, “users”, “groups”, “activity_sessions”, etc.
actions	string[]		One or more of: ‘post’, ‘put’, ‘patch’
send_to	string	<colander .required>	URL to send the webhook request to.
send_method	string	<colander .required>	One: ‘post’, ‘put’, ‘patch’
send_headers	[]{“key”: “…”, “value”: “…”}	[]	Headers attached to each request of the webhook.
signing_key	string	<colander .required>	Secret that all webhook request bodies will be signed with.

List

Return a list of this collection’s items, matching the filter parameters if provided.

HTTP Verb: `GET`

Endpoint: `/api/v1/webhooks/`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
```

```
    https://<yournamehere>.enfixlp.com/api/v1/webhooks/
```

Example Response:


```
[
{
  "_id": "5e15fe126cc4e57e33d36217",
  "endpoint": "users",
  "actions": [],
  "send_to": "http://webhook.site/asdf",
  "send_method": "post",
  "send_headers": [],
  "signing_key": "123",
  "date_added": "2020-01-08 11:06:42",
  "date_updated": "2020-01-08 11:06:42"
}
]
```


Get One

Return the specified collection item.

HTTP Verb: `GET`

Endpoint: `/api/v1/webhooks/[id]`

Example Request (cURL):

```
curl -X GET -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
    -H "Content-Type: application/json"
    https://<yournamehere>.enfixlp.com/api/v1/webhooks/[id]
```

Example Response:

```
{
  "_id": "5e15fe126cc4e57e33d36217",
  "endpoint": "users",
  "actions": [],
  "send_to": "http://webhook.site/asdf",
  "send_method": "post",
  "send_headers": [],
  "signing_key": "123",
  "date_added": "2020-01-08 11:06:42",
  "date_updated": "2020-01-08 11:06:42"
}
```


Create

Store a new instance of the object in the system. All required fields must be specified.

HTTP Verb: `POST`

Endpoint: `/api/v1/webhooks/`

Example Request (cURL):

```
curl -X POST -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     -d '{"endpoint": "users", "send_to": "http://webhook.site/asdf",
"send_method": "post", "signing_key": "123"}'
     https://<yournamehere>.enfixlp.com/api/v1/webhooks/
```

Example Response:


```
{
    "resource_url":
"https://<yournamehere>.enfixlp.com/api/v1/webhooks/51b1e78dbd7a844089b
652a9/",
```

```
    "_id": "51b1e78dbd7a844089b652a9"
}
```

Replace

This is a full model replacement of the specified item, every required field must be specified.

HTTP Verb: `PUT`

Endpoint: `/api/v1/webhooks/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     -d '{"endpoint": "users", "send_to": "http://webhook.site/asdf",
"send_method": "post", "signing_key": "123"}'
     https://<yournamehere>.enfixlp.com/api/v1/webhooks/[id]
```

Example Response:

```
No body, Status Code 204.
```

Update

On an update, you may specify only the fields you wish to update. All other fields will be preserved as currently stored.

HTTP Verb: `PATCH`

Endpoint: `/api/v1/webhooks/[id]`

Example Request (cURL):

```
curl -X PUT -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
     -H "Content-Type: application/json"
     -d '{"send_method": "post"}'
```

```
    https://<yournamehere>.enfixlp.com/api/v1/webhooks/[id]
```

Example Response:

```
No body, Status Code 204.
```

Delete

Permanently remove an item from the collection.

HTTP Verb: `DELETE`

Endpoint: `/api/v1/webhooks/[id]`

Example Request (cURL):

```
curl -X DELETE -H "App-Token: [APP_TOKEN]" -H "Token: [TOKEN]"
    -H "Content-Type: application/json"
    https://<yournamehere>.enfixlp.com/api/v1/webhooks/[id]
```

Example Response:

```
No body, Status Code 200.
```