

Wedgetail API



Wedgetail uses a RESTful interface for all API calls.

Index

[Basic Patient and Narrative Activities](#)

- [1. Creating a New Patient](#)
- [2. Adding a Narrative about a Patient](#)
- [3. Searching for a Patient](#)

[Results Service](#)

- [4. Upload a request set and ticket identifier to wedgetail](#)
- [5. Determine which request sets have been registered with wedgetail \(and therefore have an associated ticket\)](#)
- [6. Upload a set of actions and results](#)

Basic Patient and Narrative Activities

1. Creating a New Patient

Authentication via Basic HTTP Authentication

POST to <https://wedgetail.org.au/patients>

Note that new patient records will not be accessible (will remain 'unhatched') till their consent form is received by the Big Wedgie and details are confirmed.(hatched)

XML format

```
<patient>
  <family_name>Potter</family_name>
  <given_names>Harry</given_names>
  <known_as></known_as>
  <address_line>Hogwarts School</address_line>
  <town>Hogwarts</town>
  <state>NSW</state>
  <postcode>2477</postcode>
  <sex>1</sex> (1=Male 2=Female)
  <dob>1994-11-01</dob>
  <medicare></medicare>
  <dva></dva>
  <password>passwd</password>
</patient>
```

Wedgetail will return either error message or the new wedgetail number as string.

CURL example

```
curl -H 'Accept: application/xml' --user 'username:password' -H 'Content-type:application/xml' -k https://wedgetail.org.au/patients -d "<patient><family_name>Lembke</family_name><given_names>Will</given_names><known_as>Will Will</known_as><address_line>81 Dees Lane</address_line><town>Uralba</town><state>NSW</state><postcode>2477</postcode><sex>1</sex><dob>1999-02-09</dob><medicare></medicare><dva></dva><password>oddie</password></patient>" -X POST
```

2. Adding a Narrative about a patient

Authentication via Basic HTTP Authentication

Narratives are linked to patients via the patient's unique wedgetail number.

POST to <https://wedgetail.org.au/narratives.xml> (using Basic HTTP authentication (username and password))

XML Format

```
<narrative>
  <wedgetail>Hdemoc4z7v</wedgetail>
  <narrative_type_id>1 </narrative_type_id> (see below for Type Index)
  <title>Health Summary</title>
  <content>This is where the content goes as text or rtf</content>
  <content_type>text/plain or text/rtf</content_type>
  <narrative_date>2009-04-05</narrative_date>
</narrative>
```

Wedgetail API

Our current Narrative Types are >-

- 1 Health Summary
- 2 Medication Chart
- 3 Discharge Summary
- 4 Encounter
- 5 Allergies
- 6 Scratchpad
- 7 Results
- 8 Letter
- 9 Immunisations
- 10 Form
- 11 Prescription
- 12 Diagnosis
- 13 Care Plan

Curl Example

```
curl -H 'Accept: application/xml' --user 'username:password' -H 'Content-type:application/xml' -k https://wedgetail.medicine.net.au/narratives -d "<narrative><wedgetail>Hdemowua6w</wedgetail><narrative_type_id>2</narrative_type_id><title>Health Summary</title><content>Amoxil 500mg tds </content><content_type>text/plain</content_type><narrative_date>2009-08-01</narrative_date></narrative>" -X POST
```

3. Searching for a Patient

GET to <https://wedgetail.org.au/patients.xml>

Authentication via Basic HTTP Authentication

Optional Search parameters are family_name, given_names, dob (format:1999-09-04)

Note that only details about patients that the user is authorised to see will be returned.

egs

https://wedgetail.org.au/patients.xml?family_name=weasley&dob=1994-03-31

https://wedgetail.org.au/patients.xml?family_name=weasley

returns

```
<?xml version="1.0" encoding="UTF-8"?>
<patients>
  <patient>
    <family_name>Weasley</family_name>
    <given_names>Ron</given_names>
    <dob>1994-03-31</dob>
    <wedgetail>Hdemosk56y</wedgetail>
  </patient>
  <patient>
    <family_name>Weasley</family_name>
    <given_names>Ginny</given_names>
    <dob>1996-03-02</dob>
    <wedgetail>Hdemo75wvc</wedgetail>
  </patient>
</patients>
```

Curl Example

```
curl -H 'Accept: application/xml' --user 'username:password' -H 'Content-type:application/xml' -k https://wedgetail.org.au/patients.xml -d "<family_name>Weasley</family_name>" -X GET
```

wedgetail.org.au

Results Service

The steps in the result service are

a) At the time of generating a pathology request for which you would like the subsequent result uploaded to Wedgetail, send an identifier for that request (request_set) to wedgetail, and receive an identifying 'ticket' in return.

The request_set identifier should be unique to that result and to that practice.

The ticket is a string that the patient will use to later retrieve their result, and is given to the patient.

b) When a new batch of results have subsequently been received and processed, check with Wedgetail to see which request sets have been registered at the time of requesting and therefore should be uploaded.

c) upload those results and actions

d) the patient enters their identifier on the server to view relevant uploaded results

NOTE: Deidentification of all information sent to Wedgetail is the responsibility of the client program that is using the service.

DO NOT send identifying features with your result uploads.

DO NOT use this service if this is unacceptable or not possible for you.

4. Upload a request set and ticket identifier to wedgetail.

POST to http://wedgetail.org.au/result_tickets

No authorisation required

At the time a pathology request is generated by the client program, the request identifier is 'registered' with wedgetail.

A ticket identifier (used by the patient) can be nominated or generated automatically by wedgetail.

An error message is returned if you allocate a ticket that is used by a different request_set.

XML

```
<result_ticket>
  <request_set>666666</request_set>
  <ticket>010102</ticket> (optional)
</result_ticket>
```

If the ticket identifier is not included, Wedgetail will return the correct ticket identifier, if the request set has already been registered, or will generate and return a new unique ticket identifier if not already registered.

The full result_ticket with ticket identifier is returned.

```
<?xml version="1.0" encoding="UTF-8"?>
<result_ticket>
  <request_set>666666</request_set>
  <ticket>010102</ticket>
</result_ticket>
```

Curl Example

```
curl -H 'Accept: application/xml' -H 'Content-type:application/xml' -k http://localhost:3000/result\_tickets -d
"<result_ticket><request_set>666666</request_set></result_ticket>" -X POST
```

5. Determine which request sets have been registered with wedgetail (and therefore have an associated ticket)

POST to http://wedgetail.org.au/result_tickets/check

No authorisation required

XML

```
<result_ticket_list>
  <request_set>213101</request_set>
  <request_set>666666</request_set>
</result_ticket_list>
```

Returns those request_sets in the request list which have been previously registered with Wedgetail.

```
<?xml version="1.0" encoding="UTF-8"?>
<result_ticket_list>
  <request_set>666666</request_set>
</result_ticket_list>
```

Curl example

```
curl -H 'Accept: application/xml' -H 'Content-type:application/xml' -k http://wedgetail.org.au/result\_tickets/check -d
"<result_ticket_list><request_set>213101</request_set><request_set>666666</request_set></result_ticket_list>" -X POST
```

6. Upload a set of actions and results

POST To <http://wedgetail.org.au/actions>

No authorisation required

XML

```
<action_list>
  <action>
    <request_set>iii</request_set>
    <name>FBC</name>
    <action_code>1</action_code>
    <comment>You can put as much text here, and it can be in plain text or rtf</comment>

    <identifier>3</identifier>
  </action>
  <action>
    <request_set>www</request_set>
    <action_code>1</action_code>
    <comment></comment>
    <identifier>3</identifier>
  </action>
</action_list>
```

Curl example

```
curl -H 'Accept: application/xml' -H 'Content-type:application/xml' -k http://wedgetail.org.au/actions -d
"<action_list><action><request_set>iii</request_set><name>FBC</name><action_code>1</action_code><comment>Get
found</comment><identifier>3</identifier></action><action><request_set>www</request_set><action_code>1</
action_code><comment>Get found</comment><identifier>3</identifier></action></action_list>" -X POST
```