Table of Contents

# How to implement timetable

## What you need before you start:

To use the Hub Testing Integration System (HITS), you'll need the following:

* A HITS testing account - apply at [info@nsip.edu.au](mailto:info@nsip.edu.au)
* This will give you access to your own testing environment
* Your testing environment will provide you with all necessary authentication and access tokens to work with the HITS API.

You'll also need a basic knowledge of SIF REST:

* Here is [some information from the SIF Association](https://www.sifassociation.org/Resources/Developer-Resources/SIF-3-0/Pages/SIF-3.0-Infrastructure.aspx).
* Here is [a short (1-day) course in developing with SIF 3 REST](http://kb.nsip.edu.au/display/SATWI/SIF+3+Bootcamp+online).

You need to know [how to work with a usecase in HITS and access the HITS API](working_use_case_hits.md)

If you get stuck: drop us a line at [info@nsip.edu.au](mailto:info@nsip.edu.au)

## How to implement third party timetable provision

### 1. What's the business problem?

Allow schools to securely provide SIS information to the Timetabling product of their choice, and also to allow timetable records (including teaching groups and scheduled activities) to be published to a jurisdictional data hub.

[**More...**](#what-business-problem-does-this-usecase-address)

### 2. Usecase description & pre-conditions

A third party timetabling application connects to HITS as a jurisdiction hub, collecting the relevant information and publishing back timetable records to the centralised system.

#### Assumptions:

A third party vendor is a current supplier of a timetabling product in schools or has knowledge of timetabling and scheduling processes in K-12 Schools.

#### Pre-Conditions:

* Vendor has access to HITS
* HITS has been provisioned with school data
* Vendor has mapped the relevant SIF Objects to their systems:
  + StudentPersonal
  + StudentSchoolEnrolment
  + StaffPersonal
  + StaffAssignment
  + RoomInfo
  + SchoolCourseInfo (optional)
  + TimeTableSubject
  + TimeTable
  + TimeTableCell
  + TeachingGroup
  + ScheduledActivity (optional)

Here is more information on the [**Timetable Baseline Profile**](resources/Timetable%20Baseline%20Profile%20v011.pdf).

#### Usecase workflow summary:

1. Join
2. Consume
3. Process
4. Provide
5. Assurance

#### Assurance:

The SIF/XML data sent by the third party app to the jurisdiction zone for the app must be able to respond to requests for all the following records within a school:

* TimeTable
* TimeTableCell
* TeachingGroup
* ScheduledActivity (optional)

The SchoolCourseInfo object is only required if the Timetabling application will record the link between the School Course offering (as distinct from the Timetable Subject) and the Teaching Group. This link is optional, and application developers should check with the School Authority on whether they need to provide it.

The ScheduledActivity object is designed for student activities that do not involve classroom instruction, and/or that are not recurring; e.g. excursions. They do not need to be implemented by a TimeTablingApplication (and in fact they relate to scheduling rather than timetabling); application developers should check with the school authority on whether they need to provide it.

[**More...**](#usecase-preconditions-for-assurance)

### 3. Join school zone.

* Third party app connects to jurisdiction-established zone for the school ("HITS Zone 1").
* Third party app authenticates to jurisdiction-established zone for the school ("HITS Zone 1 Authz").
* Jurisdiction zone authorises read access to objects in the jurisdiction zone for the school ("HITS Zone 1 Authn").

### 4. Consume base data from HITS.

Vendor-facing (pull); HITS represents the Jurisdiction and is the data source for seed information.

Consume:

On the jurisdiction-established zone for the app, third party app accesses: - all StudentPersonal records which are in a StudentSchoolEnrollment relationship with the given School RefId - all StaffPersonal records which are in a StaffAssignment relationship with the given School RefId - all RoomInfo objects linked to the given School RefId. - all TimeTableSubject objects linked to the given School RefId. - all SchoolCourseInfo objects linked to the given School RefId. (optional)

Third party app ingests the relevant SIF objects.

1. Get SchoolInfos: http://.../SchoolInfos
2. Get StudentSchoolEnrollments: http://.../StudentSchoolEnrollments
3. Get StudentPersonals: http://.../StudentPersonals (linked to school via StudentSchoolEnrollment; equivalent to http://..../SchoolInfo/{REFID}/StudentSchoolEnrollments/{REFID}/StudentPersonals).
4. Get StaffAssignments: http://.../StaffAssignments
5. Get StaffPersonals: http://.../StaffPersonals (linked to school via StaffAssignment; equivalent to http://.../SchoolInfo/{REFID}/StaffAssignments/{REFID}/StaffPersonals).
6. Get RoomInfos: http://.../RoomInfos
7. Get TimeTableSubjects: http://.../TimeTableSubjects
8. Get SchoolCourseInfos: http://.../SchoolCourseInfos

Endpoints may support additional queries for retreiving data - refer to [**Query-by-example or service paths?**](/docs/common/qbe_or_service_paths.md) for HITS guidance on queries.

### 5. Process in third party application.

The third party app uses the consumed data to produce a timetable. The definition and automation of this process is out of scope of HITs.

Steps:

* Third party app processes information and generates timetable.
* Third party app creates return timetabling objects specific to the School.

### 6. Provide authoritative data

Prior to providing:

Third party expresses return information in SIF/XML:

* Third party app connects to jurisdiction-established zone for the School ("HITS Zone 1").
* Third party app authenticates to jurisdiction-established zone for the School ("HITS Zone 1 Authz").
* Jurisdiction-established zone authorises write access to objects in the jurisdiction zone for the school ("HITS Zone 1 Authn").
* Following is provided by the third party app back to HITS;
  1. Post TimeTable to URL http://.../TimeTables
  2. Post TimeTableCell to URL http://../TimeTableCells
  3. Post TeachingGroup to URL http://../TeachingGroups
  4. Post ScheduledActivity to URL http://.../ScheduledActivitys

### 7. Self – confirm  usecase support

Validate:

1. TimeTable records
2. TimeTableCell records
3. TeachingGroup records
4. ScheduledActivity records

### More information

#### What business problem does this usecase address?

In brief:

* Allows schools secure access to SIS information.
* Allows schools to use the Timetabling product of their choice.
* Allows third party timetabling apps automated access to base information.
* Allows third party timetabling apps to publish a school's schedule to a jurisdictional hub.

Schools currently use third-party timetabling apps locally to supplement their Student Information System (SIS). The seed information for generating timetables is held in the School's SIS and usually exported locally with little security.

As jurisdictions centralise systems, third party vendors have the opportunity to seed their product/s from a quality assured data hub using automated feeds, rather than manual updates from the school. Third party vendors are also expected to provide information directly back to the centralised system through an automated feed, rather than having the information mediated through the school.

This use case shows how third party vendors can connect to a centralised data hub to securely access to the required information and publish back the timetable records to the centralised data hub.

#### Usecase preconditions for assurance

The following conditions also must be met:

* Any SIF object published by the timetabling app must be valid against the SIF-AU 1.3 schema.
* All SIF objects posted by the timetabling app must have referential integrity. Any RefId contained in the SIF object must refer either to a SIF object provisioned to the App (e.g. SchoolInfo, StudentPersonal, RoomInfo, TimeTableSubject) or to a SIF object that has also been posted by the app to the zone (e.g. TimeTableCell referring to TimeTable). This condition applies recursively to all additional SIF objects posted by the app. The test of this condition is done only when the app indicates that it has finished publishing to the zone the objects required for the test.

For the purposes of validation, a new TeachingGroup object is well-formed if:

1. All mandatory elements of the TeachingGroup object are provided.
2. The TeachingGroup contains a TeachingGroupPeriodList.
3. The TeachingGroupPeriodList references either TimeTableCellRefId, or a combination of RoomNumber, DayId, and either PeriodId or StartTime
4. The TeachingGroup must contain a StudentList. The Students referenced in the StudentList must have StudentPersonal RefIds and Names corresponding to those supplied in provisioning.
5. The TeachingGroup must contain a TeacherList. The Teachers referenced in the TeacherList must have StaffPersonal RefIds and Names corresponding to those supplied in provisioning.
6. The TeachingGroup must reference a SchoolInfo RefId.

For the purposes of validation, a new TimeTableCell object is well-formed if it satisfies the following requirements:

* All mandatory elements of the TimeTableCell object are provided
* For the purposes of validation, an update to a TeachingGroup object can be a partially populated object, containing only the updated values, rather than a complete object.
* If an update to TeachingGroup involves any element being added, changed, or deleted in TeacherList, StudentList, or TeachingGroupPeriodList, then the entire list affected must be included in the object representation in the update request.