The Authenticated API implements OAuth 2 and OpenID Connect. To write an app using the authenticated API you need the following information:

|  |  |
| --- | --- |
| **Authorisation Endpoint:** | https://auth.sbhs.net.au/authorize |
| **Token Endpoint:** | https://auth.sbhs.net.au/token |
| **API Scopes:** | * all-ro - read-only access to a student's Portal data * openid - know a user's identity (ID Token) * profile - know a user's user information (ID Token) * email - know a user's email address (ID Token) |
|  | |
| **API Resource Endpoint:** | https://student.sbhs.net.au/api/ |

Metadata - in order to describe how to use the API (either you as the programmer, or your OAuth/OpenID library) may want to know about the following:

|  |  |
| --- | --- |
| **OAuth Metadata:** | <https://auth.sbhs.net.au/.well-known/oauth-authorization-server> |
| **OIDC Metadata:** | <https://auth.sbhs.net.au/.well-known/openid-configuration> |

Additional information that may be useful:

|  |  |
| --- | --- |
| **Access Token Life:** | 1 hour |
| **Refresh Token Life:** | 90 days |

Refresh tokens are *consumed* on use: both a new refresh token and a new access token a refresh token is used.

Lastly: Student Portal API access tokens are, technically, JWTs. Don't introspect them (parse their fields for meaning). You can't rely on the format of an access token - it could change format or meaning at any time. If you need to know who the user is, then the OpenID protocol allows you to obtain an ID token which *does* contain standardised information you can rely on.

barcodenews/list.json

**Authentication:**

Authenticated

**Return Types:**

JSON

**Parameters:**

date=YYYY-MM-DD - Date to return daily notices for. If not present, today's date is used.

**Description:**

Returns the news as displayed on the attendance computers ("Barcode Stations") for a student's year group.

**Return Example:**

{"title" : {

"date" : "1970-01-01", // date information is being returned about. (Note [1])

"content" : "Monday A (Week 1)", // description of the day

},

"content": { // news content items

"current": [ // array of items pertaining to today

{ "years" : [], // array of year groups applicable (Note [2])

"content": "" // actual news content

}, { ... },

],

"future": [ // array of items displayed today but not for today

{ "years" : [], // array of year groups applicable (Note [2])

"content": "" // actual news content

}, { ... },

]

}

**Notes**

Note 1: The date parameter is suggestive only. Always check the title.date return for the actual date returned. Future dates are not available a long time in advance and instead the *most recent* date available will be used. Weekend dates will attempt to locate the following school day.

Note 2: An array of year groups. An element can be "all" meaning any year group. Information is returned for the year group the authorising user is a member of only.

calendar/days.json

**Authentication:**

Public

**Return Types:**

JSON

**Parameters:**

from=YYYY-MM-DD - Start of date range to return information about. If not present today is used  
to=YYYY-MM-DD - End of date range to return information about. If not present start date is used

**Description:**

Returns information about the school days in the given date range - what term, week, week type etc the days are

**Return Example:**

{ // collection of dates

"2014-06-02": { // collection index

"date": "2014-06-02", // date referred to (YYYY-MM-DD)

"term": "2", // term number of day. "0" out of term

"week": "6", // week number of day. Blank or 0 on non school days.

"weekType": "A", // week type ("A", "B", "C"). Blank on non school days

"dayNumber": "2" // day number in timetable cycle of day. 0 on non school days

}, ...

}

calendar/terms.json

**Authentication:**

Public

**Return Types:**

JSON

**Parameters:**

year=YYYY - Year to return information about. If not present the current school year is used.

**Description:**

Returns information about the school year - when each term starts and ends. Also returns information about public holidays and staff development days.

**Return Example:**

"terms": { // term information

"1": { // term number

"start": { // information about the start of the term

"date": "2014-01-28", // date of first day of term

"term": "1", // term number of day

"week": "1", // week number of day

"weekType": "A", // week type ("A", "B", "C") of day

"dayNumber": "2" // day number in timetable cycle of day

},

"end": { // information about the end of the term

"date": "2014-04-11", // date of the last day of term

"term": "1",

"week": "11",

"weekType": "B",

"dayNumber": 10

}

},

...

},

"publicHolidays": { // collection of public holidays

"2014-01-01": // date of public holiday

"New Year's Day", // name of public holiday

...

},

"developmentDays": { // collection of staff development days

"2014-01-28": // date of development day

"Staff Development Day", // name of development day

...

}

dailynews/list.json

**Authentication:**

Authenticated

**Return Types:**

JSON or **RSS**

**Parameters:**

date=YYYY-MM-DD - Date to return daily notices for. If not present, today's date is used.

**Description:**

Returns the daily notices for a given date. Results are not filtered for the user's user group. If you wish to do this, obtain the user's year group from details/userinfo.json and filter the results yourself.

**Return Example (JSON):**

{"date" : "0", // UNIX timestamp

"dayInfo" : {

"date" : "1970-01-01", // YYYY-MM-DD

"term" : "2", // the school term

"week" : "2", // the week of term

"weekType" : "B", // the week type

"dayNumber" : "7", // day in timetable cycle

},

"dateYesterday: "0", // previous school day. UNIX timestamp

"dateTomorrow : "0", // next school day. UNIX timestamp

"notices" : [{ // array of notices

"title" : "Funny", // title of the daily notice

"content": "It\'s <b>Not", // body of the notice (HTML encoded)

"years" : ["7", "10" ], // applicable year groups for the notice

"dates" : ["1970-01-01"], // array of dates (YYYY-MM-DD) notice will appear

"relativeWeight" : 0, // a priority indicator for the notice

"isMeeting" : 0, // Boolean. Meetings have intrinsic +1 relative weight

"meetingDate":"1970-01-01",// date of meeting (YYYY-MM-DD), null if not a meeting

"meetingTimeParsed" : "00:00:00", // if parsable meeting time converted to HH:MM:SS

"meetingTime" : "Lunch 1", // "time" the user set the meeting for (string)

"displayYears": "All Students", // a nice way of showing message applicability

"authorName" : "R Dowdell", // display name of user who created message

},

...

]

}

**Notes**

Note 1: Fields prefixed meeting are only valid for isMeeting = 1.

Note 2: Meeting time is a string. If parsable as a time, a time is stored in meetingTimeParsed.

Note 3: If sorting notices for display, notices with isMeeting = 1 should have +1 is relativeWeight applied.

Note 4: RSS format output is possible (call dailynews/list.rss). Extended notice information that cannot be represented in RSS standard fields is added in the <dailynotice> namespace.

diarycalendar/events.json

**Authentication:**

Authenticated

**Return Types:**

JSON

**Parameters:**

from=YYYY-MM-DD - Date to list events from. If not present, today's date is used.  
to=YYYY-MM-DD - Date to list events until. If not present, one week ahead of the start date is used.

**Description:**

Returns calendar event information for the authorising student for a given date range. The information returned is sourced from: **school calendar**, **Moodle course events** ("coursework"), **school assessments**, **school homework** (not currently operating) and **personal My Diary entries**.

**Return Example (JSON):**

[ // array of dates

{

"info": { // date information

"date":"2014-06-02", // date

"term":"2", // term for date

"week":"6", // term week for date

"weekType":"B", // term week ("A", "B", "C") for date

"dayNumber":6 // day number in timetable cycle for date

},

"items":[ // array of calendar events for date

{ // normalised event information

"type":"school", // type of event (Note [1])

"subtype":"", // sub-type of event (Note [1])

"subject":"Football",// subject for event (academic or sport or cocurricular)

"title":"1st XI v G" // title line for event

"time":"09:00", // time of event (if available)

"description":"", // other body content about the event

"link":"http://...", // link to view event information, if available

"data": { } // event type-specific information

},

...

]

}, ...

]

**Notes**

Note 1: The following event types exist:

|  |  |
| --- | --- |
| school | school calendar event |
| assessment | school assessment event |
| moodle | Moodle course event. Named "coursework" in My Diary. |
| personal | user's personal My Diary entry |

**Event Type: school**

Sub-types: *none*

Type-specific data:

"index": 6, // display order in calendar for this date

"user": "Football", // name of group/body event applies to. Normalised to "Subject"

"activity": "1st XI v G", // body of calendar entry. Normalised to "Title"

"venue": "GH", // location code for location of event

"displayVenue": "Hall", // location of event

"start": "09:00", // start time of event (if available). Normalised

"end": "15:00", // end time of event

"notes": "" // any other event information. Normalised to "Description"

**Event Type: assessment**

Sub-types: *one of:* assignment, class test, exam, major work, project, assessment task, oral assessment

Type-specific data:

"assessment": "10MaA Test", // name of the assessment. Normalised to "Title"

"room": "GH", // location of the assessment

displayVenue": "Hall", // location of event

"timeFrom": "09:00", // start time of event (if available). Normalised to "Start"

"timeTo": "15:00", // end time of event

"type": "Class Test" // type of assessment. Normalised to "Subtype" and "Subject"

**Event Type: moodle**

Sub-types: *set to [module]:[type], eg.*assignment:due

Type-specific data:

"modulename":"assignment", // name of Moodle module the event is for

"eventtype":"due", // name of Moodle module event type

"name": "", // name of the event. Normalised to "Title"

"id":"123456", // unique ID of event

"description": "" // HTML code body of event. Normalised to "Description"

"timestart": 0 // UNIX timestamp for start time. Normalised to "Time" (HH:MM)

"duration": 0 // length of event in seconds

"courselink": "" // HTML code link to Moodle course. Normalised to "Subject" (plain text)

**Event Type: personal**

Sub-types: *none*

Type-specific data:

"name": "", // name of the event. Normalised to "Title"

"id":"123456", // unique ID of event

"description": "" // HTML code body of event. Normalised to "Description"

"timestart": 0 // UNIX timestamp for start time. Normalised to "Time" (HH:MM)

timetable/bells.json

**Authentication:**

Public

**Return Types:**

JSON

**Parameters:**

date=YYYY-MM-DD - Date to return bell times for. If not present, today's date is used; if the last bell of today is in the past, the next school day is used.

**Description:**

Returns information about the bell times for a given date. Information includes information about the day in the timetable cycle, the progression of periods, the time of each bell and the reason for bell changes.

**Return Example:**

{"status":"OK", // "error" if no bells for this day

"bellsAltered":true, // Boolean, true if different bell times today

"bellsAlteredReason":"Sorry Assembly",

"bells": [ // array of bell time information

{"bell":"Roll Call", // name of bell

"time":"09:00"}, // time of bell (HH:MM)

{ ...

}],

"date" : "2013-05-25", // date information returned for (YYYY-MM-DD)

"day" : "Monday", // week day name of date

"term":"2", // school term of date

"week":"5", // term week number of date

"weekType":"A" // timetable cycle week of date

}

timetable/daytimetable.json

**Authentication:**

Authenticated

**Return Types:**

JSON

**Parameters:**

|  |  |
| --- | --- |
| date=YYYY-MM-DD - | Date to return bell times for. If not present, today's date is used; if the last bell of today is in the past, the next school day is used. |

**Description:**

Returns the timetable for a student for a given date. This is what displays a student's timetable on the front page of the Student Portal.

**Return Example:**

{"status":"OK", // "error" if no bells or timetable available for this day

"date": "1970-01-01", // date this data is for (YYYY-MM-DD)

"bells": [ // array of bell time information

{"bell":"RC", // ID of bell (RC/RLn/12345 etc. "RL" is "Recess/Lunch")

"time":"09:00", // time of bell (HH:MM)

"reasonShort":"", // short reason for changed belled

"reason":"", // reason for changed bells

"bellDisplay":"Roll Call" }, // display value for this bell

{ ...

}],

"timetable": // student's timetable information

{"timetable": // timetable slice for this day

{"dayname":"Tuesday A", // day display name

"routine": "0,RC,1,2...", // day routine, comma-separated list of bells/periods in order

"rollcall": // roll class info

{"title":"09M", // roll class name

"teacher":"DO", // roll teacher code

"room":"" // roll class room

},

"periods":{ // collection of periods

"1":{ // period number (aligns to routine)

"title":"Ma1", // name of class

"teacher":"BS", // teacher code

"fullTeacher":"Mr R Boros",

"room":"105", // class room

"year":"9" // year for class (Note [1])

},

...

},

"subjects":{ // collection of information about subjects

"9Ma1":{ // class lookup key (Note [2])

"title":"9 Maths 1", // full name of class

"shortTitle":"Ma1", // short name of class (see timetable)

"teacher":"AE", // class teacher code (see timetable)

"subject":"English", // class subject

"fullTeacher":"Ms S Ayre",

"year":"9", // year for class (Note [1])

"colour":"cccccc" // RGB colour code for this subject (whether it's legible is your problem)

},

...

},

}

}

"roomVariations": { // array of applicable room variations

"3": {

"period": "3", // period code for variation

"year": 9, // year group variation is for (Note [1])

"title": "Ma1", // class shortTitle for affected class

"roomFrom": "101", // obvious

"roomTo": "102" // room moved to

},

...

}

},

"classVariations": { // collection of applicable room variations

"3" : {

"period": "3", // period code for variation

"year": 9, // year group variation is for (Note [1])

"title": "Ma1", // class shortTitle for affected class

"teacher": "DW", // teacher code of absent teacher

"type": "nocover", // see Note[3]

"casual": "CX", // teacher code of replacement

"casualSurname": "Craddock" // name of replacement

},

...

}

},

"serverTimezone":"36000", // timezone (seconds from UTC) of all times showsn

"shouldDisplayVariations":false // Boolean indication of whether the variations are "fixed" for this day

}

**Notes**

Note 1: Implementers must take note of the Year returned. Accelerants may be in multiple years at once (for example 10 and 11). The higher year's classes take precedence over the lower year's (e.g. *11* Acc. Music over *10* Geography).

Note 2: The key name of each element in the subjects collection can be constructed from the period information: ie timetable.subjects[timetable.timetable.periods[x].year + timetable.timetable.periods[x].title] provides subject information for period x.

Note 3: Variation type may be one of nocover (no teacher), replacement (substitute teacher per casual field) or novariation (the regular teacher will be teaching the class).

timetable/timetable.json

**Authentication:**

Authenticated

**Return Types:**

JSON

**Parameters:**

None

**Description:**

Returns the complete timetable for a student.

**Return Example:**

{"student": {

"surname": "",

"givenname": "",

"sex": "", // "M"/"B" for male

"DOB": "0", // date of birth. Unix timestamp

"roll": "1", // index of roll class

"lines": { // collection of line indexes

"1":"5", // student does class 5 on line 1

...

}

"extraLines": { // collection of extra line indexes

"1":"5", // student does item 5 on extra line 1

...

}

"BoSNumber": "" , // Board of Studies number. 0 if not available

"studentId": "43...", // Student ID number

"year": "10", // student's [primary] year group

"years": ["10",...] // array of years the student is in

}

"days": { // collection of days in the timetable

"1" : {

"dayname": "Monday A" // name of the day

"routine": "0,RC,1,2...", // routine (see timetable/daytimetable()

"rollcall": { // roll call info for this day

"title": "07T", // name of roll class

"teacher": "DW", // teacher code for roll class

"room": "" // room for roll call

},

"periods": // collection of periods for this day

"1": { // corresponds to routine

"title": "Ma1", // short name for class

"teacher": "DW", // teacher code for class

"room": "101", // room for class

"year": "8" // year for class (Note [1])

}, ...

},

...,

},

"subjects" : { // collection of subject information

"1": {

"title": "8 Maths 1", // title of the class

"shortTitle: "Ma1", // corresponds to period.title

"subject" "Mathematics" // full name of the subject

"teacher": "DW", // teacher code for the class teacher

"fullTeacher": "Mr R Dow" // full name of the teacher

"year": "8" // year for the class (Note [1])

},

...

}

"extraSubjects": { // collection of extra subjects

"1": { // these subjects have no timetable classes

"title": "Student Advis", // title of the class

"shortTitle: "SA7", // corresponds to period.title

"teacher": "DW", // teacher code for the class teacher

"fullTeacher": "Mr R Dow" // full name of the teacher

},

...

},

"rollcall": { // roll class details

"title": "08M" , // title of the class

"teacher": "DW", // teacher code for the teacher

"fullTeacher": "Mr R Dow" // full name of the teacher

"room": " " // roll class room

},

"advisor": "Mr R Dowdell" // year adviser

}

**Notes**

Note 1: Implementers must take note of the Year returned. Accelerants may be in multiple years at once (for example 10 and 11). The higher year's classes take precedence over the lower year's (e.g. *11* Acc. Music over *10* Geography).

details/participation.json

**Authentication:**

Authenticated

**Return Types:**

JSON

**Parameters:**

None

**Description:**

Returns a student's Award Scheme participation information

**Return Example:**

[ // array of participation information

{"year" : "2014", // year of activity

"activity" : "Smiling", // name of activity

"category" : "90", // award scheme category number

"categoryName" : "Bonus", // name of category

"points" : "1", // number of points awarded

"pointsCap" : 10 // maximum number of points achievable in the category

},

...

]

details/userinfo.json

**Authentication:**

Authenticated

**Return Types:**

JSON

**Parameters:**

None

**Description:**

Returns information about the user the supplied Access Token is valid for

**Return Example:**

{"username" : "436345789",

"studentId" : "436345789",

"givenName" : "John",

"surname" : "Citizen",

"rollClass" : "07E",

"yearGroup" : "7",

"role" : "Student", // may be valid for staff

"department": "Year 7", // may be valid for staff

"office" : "7E", // may be valid for staff

"email" : "[436345789@student.sbhs.nsw.edu.au](mailto:436345789@student.sbhs.nsw.edu.au)",

"emailAliases : [ // array of email addresses also valid for the user

"[john.citizen23@student.sbhs.nsw.edu.au](mailto:john.citizen23@student.sbhs.nsw.edu.au)"],

"decEmail" : "[jcz@education.nsw.gov.au](mailto:jcz@education.nsw.gov.au)",

"groups" : [] // array of network group memberships

}