



Altora REST API v1.09a

The API

The Altora REST API is accessed using the following endpoint URL:

<https://api.userlogin.com.au>

Authentication:

All requests must be sent with your 'x-api-key' and 'x-api-secret' in the HTTP header fields.

Resources Available

The GET, POST, PATCH and PUT resources available are detailed in the following headings.

The {clientid} part of the resource URI is your clientid identifier. This can be found by looking at the URL that you normally use to access your Altora online inductions system. If the URL you use is <https://www.userlogin.com.au/yourcompany>, then your {clientid} is yourcompany.

GET /v1/{clientid}/users/{userid}

Get a user's details

Request Information

Parameters

Name	Description	
Userid	A User object's ID	(Required)

Response Information

Returns

A user's basic details.

Response Body Format

On Success:

```
{
  "Id": "1234",
  "Firstname": "John",
  "Lastname": "Citizen",
  "Mobile": "+61401234567",
  "Phone": "(03)90000000",
  "Position": "Consultant",
  "Email": "person@email.com.au",
  "Company_Name": "Acme Inc",
  "Dob": "1997-08-27",
  "Active": "1",
  "Username": "1234"
}
```

On Failure:

Example response if user does not exist:

```
{
  "Status": "error",
  "Message": "User with id 10049 does not exist!"
}
```

POST /v1/{clientid}/users

Add a new user

Request Information

Request Body format

Include data as a JSON object in the request body. Set “Content-Type” header to “application/json”.

Name	Description	
Firstname	User’s Firstname	Required.
Lastname	User’s Lastname	Required.
Mobile	User’s Mobile number	
Phone	User’s Phone number	
Email	User’s Email	
Company_Name	User’s Company name	
Position	User’s Position	
Dob	User’s date of birth	

```
{
  "Firstname": "Jane",
  "Lastname": "Citizen",
  "Mobile": "+61412345678",
  "Phone": "(03) 99991234",
  "Email": "person@email.com.au",
  "Company_Name": "ACME Inc",
  "Position": "Branch Manager",
  "Dob": "1990-01-15"
}
```

Response Information

Returns

System generated Username and Password of the created user.

Response Body Format

```
{
  "status": "success",
  "username": "10100",
  "password": "abcd123"
}
```

PATCH /v1/{clientid}/users/{userid}

Update a user's details

Request Information

Parameters

Name	Description	
userid	A User object's ID	Required.

Request Body format

Include data as a JSON object in the request body. Set "Content-Type" header to "application/json". Only include fields that are to be updated. (Fields that remain the same can be omitted)

Name	Description	
Firstname	User's firstname	
Lastname	User's lastname	
Mobile	User's Mobile number	
Phone	User's Phone number	
Position	User's position	
Email	User's email address	
Dob	User's date of birth	
Active	User's active status	
Company_Name	User's Company Name (Used if Company module is not enabled)	
CompanyId	User's Company Object Id (Used if Company module is enabled)	

```
{
  "Firstname": "John",
  "Lastname": "Citizen",
  "Mobile": "+61401234567",
  "Phone": "0390000000",
  "Position": "Consultant",
  "Email": "person@email.com.au",
  "Dob": "1997-08-27",
  "Active": "1",
  "CompanyId": "2"
}
```

Response Information

Returns

Success or Error.

Response Body Format

On Success:

```
{  
  "status": "success",  
  "message": "User with Id 10003 updated."  
}
```

On Failure:

Example if user does not exist:

```
{  
  "Status": "error",  
  "Message": "User does not exist!"  
}
```

GET /v1/{clientid}/courses/

Get a list of all available courses and their object ids.

Response Information

Returns

A list of all courses.

Response Body Format

```
[
  {
    "Id": "1",
    "Course_Name": "First Course"
  },
  {
    "Id": "2",
    "Course_Name": "Second Course"
  },
  {
    "Id": "3",
    "Course_Name": "Third Course"
  }
]
```

GET /v1/{clientid}/documents/

Get a list of all available documents and corresponding object ids. (Documents are uploaded documents such as Licences/Insurances or Photos etc.)

Response Information

Returns

A list of all documents with their object ids.

Response Body Format

```
[
  {
    "Id": "1",
    "Document_Name": "First Document"
  },
  {
    "Id": "2",
    "Document_Name": "Second Document"
  },
  {
    "Id": "3",
    "Document_Name": "Third Document"
  }
]
```

GET /v1/{clientid}/forms/

Get a list of all available forms and corresponding object ids.

Response Information

Returns

A list of all forms with their object ids.

Response Body Format

```
[
  {
    "Id": "1",
    "Form_Name": "First Form"
  },
  {
    "Id": "2",
    "Form_Name": "Second Form"
  },
  {
    "Id": "3",
    "Form_Name": "Third Form"
  }
]
```


GET /v1/{clientid}/acknowledgements/

Get a list of all available acknowledgements and their object ids.

Response Information

Returns

A list of all acknowledgements in the system with their object ids.

Response Body Format

```
[
  {
    "Id": "1",
    "Acknowledgement_Name": "First Acknowledgement"
  },
  {
    "Id": "2",
    "Acknowledgement_Name": "Second Acknowledgement"
  },
  {
    "Id": "3",
    "Acknowledgement_Name": "Third Acknowledgement"
  }
]
```

GET /v1/{clientid}/scorm-courses/

Get a list of all available SCORM courses and their object ids and SCORM course name.

Response Information

Returns

A list of all SCORM courses.

Response Body Format

```
[
  {
    "Id": "1",
    "SCORM_Name": "First SCORM Course"
  },
  {
    "Id": "2",
    "SCORM_Name": "Second SCORM Course"
  },
  {
    "Id": "3",
    "SCORM_Name": "Third SCORM Course"
  }
]
```

GET /v1/{clientid}/programs/

Get a list of all available programs and corresponding object ids.

Response Information

Returns

A list of all programs with their object ids.

Response Body Format

```
[
  {
    "Id": "36",
    "Program_Name": "Program A"
  },
  {
    "Id": "37",
    "Program_Name": "Program B"
  },
  {
    "Id": "38",
    "Program_Name": "Program C"
  }
]
```

GET /v1/{clientid}/users/{userid}/documents?documentid={documentid}

Get document data submitted by a user. (Licence/Insurance/Photo etc.)

Request Information

Parameters

Name	Description	
Userid	A user's object Id	(Required)
Documentid	A Document object's Id	(Required)

Response Information

Returns

Document data last submitted by User of a specific document ID.

If the document contains an issue date or expiry date, they will be shown.

If the document data contains an uploaded file (such as a photo), a time restricted secure link is provided to download the file.

If the document contains an Issuer field, the field will be provided.

Response Body Format

```
{
  "Issue_Date": "2015-11-01 05:02:57",
  "Expiry_Date": "2016-08-02 05:02:57",
  "Document_Number": "A1234567",
  "File": "https://link.to/file",
  "Issuer": "ABC Training Company",
}
```

GET

/v1/{clientid}/users/{userid}/acknowledgements?acknowledgementid={acknowledgementid}

Get acknowledgement data submitted by a user. (Whether user has submitted an acknowledgment and/or signed an acknowledgement)

Request Information

Parameters

Name	Description	
Userid	A user object's Id	(Required)
Acknowledgementid	An acknowledgement object's Id	(Required)

Response Information

Returns

Acknowledgement data last submitted by a User for a specific Acknowledgement.

If the acknowledgement contains a signature, a time restricted secure link is provided to download the signature as an image.

Response Body Format

```
{
  "Acknowledgement_Number": "123456789",
  "File": "https://link.to/file",
  "Acknowledged_By": "John Smith",
  "Time_Acknowledged": "2019-02-01 13:24:00"
}
```

GET /v1/{clientid}/users/{userid}/forms?formid={formid}

Get latest form data submitted by a user.

Request Information

Parameters

Name	Description	
Userid	User object's Id	(Required)
Formid	Form object's Id	(Required)

Response Information

Returns

Latest occurrence of form submitted by user.

Data returned will be Form Field Labels and Values with Headings.

Response Body Format

```
{
  "Fields": [
    {
      "label": "",
      "value": "This is a Heading"
    },
    {
      "label": "Field Name 1",
      "value": "Value 1"
    },
    {
      "label": "Field Name 2",
      "value": "Value 2"
    },
    {
      "label": "",
      "value": "This is a Heading 2"
    },
    {
      "label": "Field Name 3",
      "value": "Value 3"
    },
    {
      "label": "Field Name 4",
      "value": "Value 4"
    }
  ]
}
```

GET /v1/{clientid}/users/{userid}/courses?courseid={courseid}

Get courses previously completed by a user.

Request Information

Parameters

Name	Description	
Userid	User object's Id	(Required)
Courseid	Course object's Id	(Optional) Provide course id to filter results to one specific course .

Response Information

Returns

Completed course details and expiry dates if applicable.

Response Body Format

```
[
  {
    "Course_Id": "70",
    "Course_Name": "Course Name A",
    "Issue_Date": "2015-02-10",
    "Expiry_Date": "2016-08-02",
    "Certificate_No": "000054"
  },
  {
    "Course_Id": "71",
    "Course_Name": "Course Name B",
    "Issue_Date": "2015-05-05",
    "Expiry_Date": "2016-05-05",
    "Certificate_No": "000060"
  },
  {
    "Course_Id": "70",
    "Course_Name": "Course Name A",
    "Issue_Date": "2015-05-05",
    "Expiry_Date": "2016-07-05",
    "Certificate_No": "000061"
  },
  {
    "Course_Id": "72",
    "Course_Name": "Course Name C (Non-expiring)",
    "Issue_Date": "2015-05-22",
    "Expiry_Date": "",
    "Certificate_No": "000078"
  }
]
```

GET /v1/{clientid}/users/{userid}/scorm-courses?scormid={scormid}

Get SCORM courses previously completed by a user.

Request Information

Parameters

Name	Description	
userid	User's object Id	(Required)
scormid	SCORM Course's object Id	(Optional) Provide scorm id to filter results to one specific SCORM course .

Response Information

Returns

Completed course details and expiry dates if applicable.

Response Body Format

```
[
  {
    "SCORM_Id": "70",
    "SCORM_Name": "SCORM Course Name A",
    "Issue_Date": "2025-02-10",
    "Expiry_Date": "2026-08-02",
    "Certificate_No": "000054"
  },
  {
    "SCORM_Id": "71",
    "SCORM_Name": "SCORM Course Name B",
    "Issue_Date": "2025-05-05",
    "Expiry_Date": "2026-05-05",
    "Certificate_No": "000060"
  },
  {
    "SCORM_Id": "70",
    "SCORM_Name": "SCORM Course Name A",
    "Issue_Date": "2025-05-05",
    "Expiry_Date": "2026-07-05",
    "Certificate_No": "000061"
  },
  {
    "SCORM_Id": "72",
    "SCORM_Name": "SCORM Course Name C (Non-expiring)",
    "Issue_Date": "2025-05-22",
    "Expiry_Date": "",
    "Certificate_No": "000078"
  }
]
```


GET /v1/{clientid}/users/{userid}/programs

Get all programs assigned to the specified user.

Request Information

Parameters

Name	Description	
Userid	User object's Id	(Required)

Response Information

Returns

All programs allocated to the specified user represented as an array of Program Ids.

Response Body Format

```
{
  "Programs": [ 1, 2, 3]
}
```

PUT /v1/{clientid}/users/{userid}/programs

Set assigned programs for the specified user.

Request Information

Parameters

Name	Description	
Userid	User object's Id	(Required)

Request Body format

Include data as a JSON object in the request body. Set "Content-Type" header to "application/json".

Name	Description	
Programs	An array containing the complete set of Program Ids the user is assigned to.	Required.

Example:

```
{
  "Programs": [ 2, 3, 5 ]
}
```

Response Information

Returns

Status message on whether the assignment was successful.

Response Body Format

On Success:

```
{
  "Status": "success",
  "Message": "Program assignment completed successfully."
}
```

On Failure:

Example if user does not exist:

```
{
  "Status": "error",
  "Message": "User does not exist!"
}
```

GET

/v1/{clientid}/users?firstname={firstname}&lastname={lastname}&email={email}

Get/Search for details of one or more users that match the supplied parameters.

If no parameters are supplied, all users will be returned.

Request Information

Parameters

Name	Description	
Firstname	User's firstname.	(Optional)
Lastname	User's lastname.	(Optional)
Email	User's email address	(Optional)

Response Information

Returns

Details of all users that match the supplied parameters.

Response Body Format

```
[
  {
    "Id": "1234",
    "Firstname": "John",
    "Lastname": "Smith",
    "Mobile": "+61400123456",
    "Phone": "0390001111",
    "Position": "Analyst",
    "Email": "john.smith@companya.com.au",
    "Company_Name": "Company A",
    "Dob": "1970-09-30",
    "Active": "1",
    "Username": "1234"
  },
  {
    "Id": "1235",
    "Firstname": "John",
    "Lastname": "Brown",
    "Mobile": "+61411222333",
    "Phone": "0398761234",
    "Position": "Manager",
    "Email": "john.brown@companyb.com.au",
    "Company_Name": "Company B",
    "Dob": "1975-01-15",
    "Active": "1",
    "Username": "1235"
  }
]
```

GET

/v1/{clientid}/users?firstname={firstname}&lastname={lastname}&email={email}&offset={offset}&limit={limit}

Paginated version of the endpoint to Get/Search for details of one or more users that match the supplied parameters.

If no search parameters are supplied, all users will be returned.

Request Information

Parameters

Name	Description	
Firstname	User's firstname.	(Optional)
Lastname	User's lastname.	(Optional)
Email	User's email address	(Optional)
Limit	The maximum amount of results to return	
Offset	The amount of offset to the start of the results	

When a {limit} is provided, the results returned by the Users endpoint will be paginated. A maximum amount of results up to the {limit} is returned starting from the {offset}.

If {limit} is not supplied, the standard unpaginated Response Body format is returned.

Paginated Response Body

A previous and next URL will be provided in the returned results to access the next or previous page of results for convenience. If a "next" URL is not provided, the last page of the results have been reached. If a "prev" URL is not provided, the first page of results have been reached.

The user data will be returned under "results".

```
{
  "pagination": {
    "next":
"https://api.userlogin.com.au/v1/clientid/users?offset=10000&limit=5000",
    "prev":
"https://api.userlogin.com.au/v1/clientid/users?offset=0&limit=5000"
  },
  "results": [
    {
      "Id": "10123",
      "Firstname": "John",
      "Lastname": "Smith",
      "Mobile": "+61401122333",
      "Phone": "(03) 90001111",
      "Position": "",
      "Email": john.smith@acmeinc.com,
      "Company_Name": "Acme Inc",
      "Dob": null,
    }
  ]
}
```

```

        "Active": "1",
        "Username": "10123"
    },
    {
        "Id": "10124",
        "Firstname": "Sam",
        "Lastname": "Jones",
        "Mobile": "",
        "Phone": "",
        "Position": "",
        "Email": "Sam.Jones@acmeinc.com",
        "Company_Name": "Acme Inc",
        "Dob": null,
        "Active": "1",
        "Username": "10124"
    }
]
}

```

For example, to get the complete set of users, first call:

<https://api.userlogin.com.au/v1/clientid/users?offset=0&limit=5000>

and then: <https://api.userlogin.com.au/v1/clientid/users?offset=5000&limit=5000>

and then <https://api.userlogin.com.au/v1/clientid/users?offset=10000&limit=5000>

and so on until the “next” URL is no longer provided in the returned results.

GET /v1/{clientid}/reports/completed-courses/{courseid}?fromdate={fromdate}&todate={todate}

Get a list of completed courses between two set dates.

Request Information

Parameters

Name	Description	
Courseid	A Course object's ID	(Required)
Fromdate	Date to run report from	(Required) Format: dd/mm/yyyy
Todate	Date to run report to	(Required) Format: dd/mm/yyyy

Response Information

Returns

User ID, date user was inducted and when the completed course expires for user.

Response Body Format

```
[
  {
    "User_Id": "1409",
    "Issue_Date": "29-10-2015",
    "Expiry_Date": "29-10-2016"
  },
  {
    "User_Id": "1792",
    "Issue_Date": "04-11-2015",
    "Expiry_Date": "04-11-2016"
  },
  {
    "User_Id": "1",
    "Issue_Date": "10-11-2015",
    "Expiry_Date": "10-11-2016"
  }
]
```

GET /v1/{clientid}/reports/completed-scorm-courses?fromdate={fromdate}&todate={todate}&scormid={scormid}

Get a list of completed SCORM courses between two set dates.

Request Information

Parameters

Name	Description	
fromdate	Date to run report from	(Required) Format: yyyy-mm-dd
todate	Date to run report to	(Required) Format: yyyy-mm-dd
scormid	A SCORM Course object's ID	(Optional) If supplied, results will be filtered by SCORM course Id.

Response Information

Returns

SCORM Id, User Id, date the user completed SCORM course (issue date) and expiry date of the SCORM course for user.

Response Body Format

```
[
  {
    "SCORM_Id": "43",
    "User_Id": "10411",
    "Issue_Date": "2025-10-01",
    "Expiry_Date": "2027-10-01"
  },
  {
    "SCORM_Id": "123",
    "User_Id": "10072",
    "Issue_Date": "2025-12-05",
    "Expiry_Date": "2026-12-05"
  },
  {
    "SCORM_Id": "87",
    "User_Id": "10034",
    "Issue_Date": "2024-11-15",
    "Expiry_Date": "2025-11-15"
  }
]
```

GET /v1/{clientid}/company-tasks

Get list of all company tasks available

Response Information

Returns

An array of all available company tasks

Response Body Format

```
[
  {
    "Id": "1",
    "Task_Name": "Public Liability Insurance",
    "Type": "insurance"
  },
  {
    "Id": "2",
    "Task_Name": "Registered Electrical Contractor",
    "Type": "licence"
  },
  {
    "Id": "3",
    "Task_Name": "Safety Management Plan",
    "Type": "swms"
  }
]
```


GET /v1/{clientid}/companies/

Get a list of all available companies and their object ids.

Response Information

Returns

A list of all companies.

Response Body Format

```
[
  {
    "Id": "1",
    "Trading_Name": "Acme Inc",
    "Active": "1"
  },
  {
    "Id": "2",
    "Trading_Name": "Company ABC",
    "Active": "1"
  },
  {
    "Id": "3",
    "Trading_Name": "Altora Pty Ltd",
    "Active": "1"
  }
]
```

POST /v1/{clientid}/companies/

Create a new company.

Request Body format

Include data as a JSON object in the request body. Set “Content-Type” header to “application/json”.

Name	Description	
Trading_Name	Company’s Trading Name	Required.
ABN	Company’s ABN	
Phone	Company’s Phone Number	
Business_Structure	Company’s Business Structure: “Company”, “Sole trader”, “Partnership”, “Trust” or “Other”	
Active	Company’s Active Status	
Business_Address	Object with the fields below:	
Street	Business Address Street	
Suburb	Business Address Suburb	
State	Business Address State: “Victoria”, “New South Wales”, “Australian Capital Territory”, “Queensland”, “Northern Territory”, “South Australia”, “Tasmania”, “Western Australia” or “Outside of Australia”	
Postcode	Postcode	
Postal_Address	Object with the fields below:	
Street	Postal Address Street	
Suburb	Postal Address Suburb	
State	Postal Address State	
Postcode	Postal Address Postcode	

```
{
  "Trading_Name": "Acme Inc",
  "ABN": "68153003368",
  "Phone": "(03) 90001234",
  "Business_Structure": "Company",
  "Active": 1,
  "Business_Address": {
    "Street": "1 Main Street",
    "Suburb": "Melbourne",
    "State": "Victoria",
    "Postcode": "3000"},
  "Postal_Address": {
    "Street": "123 Postal Street",
    "Suburb": "Sydney",
    "State": "New South Wales",
    "Postcode": "2000"
  }
}
```

```
}  
}
```

Response Information

Returns

Status and companyId of the newly added Company if successful.

Response Body Format

```
{  
  "status": "success",  
  "companyId": "57"  
}
```

GET /v1/{clientid}/companies/{companyid}/

Get the company details for a specific company.

Request Information

Parameters

Name	Description	
companyid	Company object's Id	(Required)

Response Information

Returns

A company's details

Response Body Format

```
{
  "Id": "2",
  "Trading_Name": "Acme Inc",
  "Active": "1",
  "Entity_Name": "",
  "ABN": "32051954940",
  "Phone": "+61312345678",
  "Business_Structure": "Company",
  "Created_Date": "2019-11-28 02:38:26",
  "Business_Address": {
    "Street": "1 Main Street",
    "Suburb": "Melbourne",
    "State": "Victoria",
    "Postcode": "3000"
  },
  "Postal_Address": {
    "Street": "1 Postal Street",
    "Suburb": "Not Melbourne",
    "State": "New South Wales",
    "Postcode": "2000"
  }
}
```

PATCH /v1/{clientid}/companies/{companyid}/

Update the company details for a specific company.

Request Information

Parameters

Name	Description	
companyid	Company object's Id	(Required)

Request Body Format

Include data as a JSON object in the request body. Set "Content-Type" header to "application/json". Only includes fields that need to be updated. (Fields that do not need to be updated can be omitted)

Name	Description	
Id	Company's Object Id	Required.
Trading_Name	Company's Trading Name	
ABN	Company's ABN	
Phone	Company's Phone Number	
Business_Structure	Company's Business Structure	Must be one of the following: "Company", "Sole trader", "Partnership", "Trust" or "Other"
Active	Company's Active Status	
Business_Address	Object with the fields below:	
Street	Business Address Street	
Suburb	Business Address Suburb	
State	Business Address State	Must be one of the following: "Victoria", "New South Wales", "Australian Capital Territory", "Queensland", "Northern Territory", "South Australia", "Tasmania", "Western Australia" or "Outside of Australia"
Postcode	Postcode	
Postal_Address	Object with the fields below:	
Street	Postal Address Street	
Suburb	Postal Address Suburb	
State	Postal Address State	
Postcode	Postal Address Postcode	

```
{
  "Trading_Name": "My Trading Name",
  "ABN": "123456789",
  "Phone": "",
  "Business_Structure": "Company",
  "Active": 1,
  "Business_Address": {
    "Street": "1 Main Street",
    "Suburb": "Melbourne",
    "State": "Victoria",
    "Postcode": "3000"},
  "Postal_Address": {
    "Street": "1 Postal Street",
    "Suburb": "Not Melbourne",
    "State": "Victoria",
    "Postcode": "2000"
  }
}
```

Response Information

Returns

Success or Failure.

Response Body Format

```
{
  "status": "success"
  "message": "Company with id 52 updated."
}
```

GET /v1/{clientid}/companies/{companyid}/company-tasks

Get the company tasks allocated to a company.

Request Information

Parameters

Name	Description	
companyid	Company object's Id	(Required)

Response Information

Returns

Tasks field containing an array of company tasks currently allocated to this company.

Response Body Format

```
{ "Tasks": [1, 2, 3] }
```

PUT /v1/{clientid}/companies/{companyid}/company-tasks

Allocate company tasks to a company

Request Information

Parameters

Name	Description	
companyid	Company object's Id	(Required)

Request Body format

Include data as a JSON object in the request body. Set "Content-Type" header to "application/json".

Name	Description	
Tasks	An array containing the complete set of Tasks Ids to be assigned to this company.	Required.

```
{
  "Tasks": [1, 2, 3]
}
```

Response Information

Returns

Success or Failure.

Response Body Format

On Success:

```
{
  "Status": "success",
  "Message": "Company Task assignment completed successfully."
}
```

On Failure:

```
{
  "Status": "error",
  "Message": "Invalid task ids supplied! - 521"
}
```


GET /v1/{clientid}/companies/{companyid}/admins

Get the User ids of the user accounts that are the currently assigned company admins for a company.

Request Information

Parameters

Name	Description	
companyid	Company object's Id	(Required)

Response Information

Returns

An array containing the user ids of the user accounts that are currently a company admin for the company. Only users who have accepted an admin invitation will be included.

Response Body Format

```
[
  {
    "userId": "10929"
  },
  {
    "userId": "10948"
  }
]
```

PUT /v1/{clientid}/companies/{companyid}/admins/invite

Invite an existing user to be a company admin for a specific company.

Request Information

Parameters

Name	Description	
companyid	Company object's Id	(Required)

Request Body format

Include data as a JSON object in the request body. Set "Content-Type" header to "application/json".

Include userId of the user to be invited to the company as a company admin.

Parameters

Name	Description	
userId	User object's Id of the User to be invited to be an admin for this company	(Required)

Example:

```
{  
  "userId": "10929"  
}
```

Response Body Format

On Success:

```
{  
  "Status": "success",  
  "Message": "Company active value saved successfully."  
}
```

On Failure:

Example if company does not exist:

```
{  
  "Status": "error",  
  "Message": "Company does not exist!"  
}
```

GET /v1/{clientid}/sites

Get a list of all available sites

Response Information

Returns

A list of sites.

Response Body Format

```
[
  {
    "id": "1",
    "name": "Building Site",
    "description": "This site is located on Main Rd",
    "timeout": "24",
    "preferences":
    {
      "prevent-invalid": "false",
      "notify-check-in": "false",
      "notify-failure": "false",
      "display-on-site": "false",
      "default-check-in-type": "mobile",
      "allow-visitors": "false"
    },
    "created": "2022-10-26 23:44:40"
  },
  {
    "id": "2",
    "name": "Mining Site",
    "description": "Main mining site",
    "timeout": "8",
    "preferences":
    {
      "prevent-invalid": "true",
      "notify-check-in": "true",
      "notify-failure": "true",
      "display-on-site": "false",
      "default-check-in-type": "mobile",
      "allow-visitors": "false"
    },
    "created": "2023-04-11 05:34:04"
  }...
]
```

GET /v1/{clientid}/sites/{id}

Get a site's details

Request Information

Parameters

Name	Description	
id	A Site object's ID	Required

Response Information

Returns

A site's basic details.

Response Body Format

Preferences are all true/false expect default-check-in-type which accepts mobile or username.

On Success:

```
{
  "id": "1",
  "name": "Building Site",
  "description": "This site is located on Main Rd",
  "timeout": "24",
  "preferences": {
    "prevent-invalid": "false",
    "notify-check-in": "false",
    "notify-failure": "false",
    "display-on-site": "false",
    "default-check-in-type": "mobile",
    "allow-visitors": "false"
  },
  "created": "2022-10-26 23:44:40"
}
```

On Failure:

Example response if site does not exist:

```
{
  "status": "error",
  "message": "Site not found"
}
```

POST /v1/{clientid}/sites

Create a new site.

Request Body format

Include data as a JSON object in the request body. Set “Content-Type” header to “application/json”.

Name	Description	
name	Site’s name	Required
description	Describes the site	
timeout	Time in hours (integer); maximum time check in remains on site (0 is no timeout)	
preferences	Object with the fields below:	
prevent-invalid	Stop workers from checking in if they haven’t completed the necessary requirements	Must be true or false
notify-check-in	Notify site administrators when successful check in	Must be true or false
notify-failure	Notify site administrators when worker fails to meet check in requirements	Must be true or false
display-on-site	Allow kiosk to display users on site	Must be true or false
default-check-in-type	Identifier that majority of workers check in with	Only options supported are mobile or username
allow-visitors	Allow un-registered users to check-in as a visitor	Must be true or false

```
{
  "name": "Site Name",
  "description": "This is our site",
  "timeout": "8",
  "preferences": {
    "prevent-invalid": "true",
    "notify-check-in": "true",
    "notify-failure": "false",
    "display-on-site": "false",
    "default-check-in-type": "mobile",
    "allow-visitors": "false"
  }
}
```

Response Information

Returns

Status and id of the newly added site if successful.

Response Body Format

On Success:

```
{  
  "status": "success",  
  "message": "Site created",  
  "siteId": "44"  
}
```

On Failure:

Example response if site wasn't created:

```
{  
  "status": "error",  
  "message": "Failed to create site"  
}
```

PATCH /v1/{clientid}/sites/{id}

Update the details or settings for a single site.

Request Information

Parameters

Name	Description	
id	Site object's Id	Required

Request Body Format

Include data as a JSON object in the request body. Set "Content-Type" header to "application/json". Only includes fields that need to be updated. (Fields that do not need to be updated can be omitted)

Name	Description	
name	Site's name	
description	Describes the site	
timeout	Time in hours (integer); maximum time check in remains on site (0 is no timeout)	
preferences	Object with the fields below:	
prevent-invalid	Stop workers from checking in if they haven't completed the necessary requirements	Must be true or false
notify-check-in	Notify site administrators when successful check in	Must be true or false
notify-failure	Notify site administrators when worker fails to meet check in requirements	Must be true or false
display-on-site	Allow kiosk to display users on site	Must be true or false
default-check-in-type	Identifier that majority of workers check in with	Only options supported are mobile or username
allow-visitors	Allow un-registered users to check-in as a visitor	Must be true or false

```
{
  "name": "Site Name",
  "description": "This is our site",
  "timeout": "8",
  "preferences": {
    "prevent-invalid": "true",
    "notify-check-in": "true",
    "notify-failure": "false",
    "display-on-site": "false",
    "default-check-in-type": "mobile",
    "allow-visitors": "false"
  }
}
```

Response Information

Returns

Success or Failure.

Response Body Format

On Success:

```
{  
  "status": "success"  
  "message": "Site updated."  
}
```

On Failure:

Example response if site wasn't created:

```
{  
  "status": "error",  
  "message": "Failed to update site"  
}
```


GET /v1/{clientid}/sites/{id}/terms

Get a site's terms and conditions

Request Information

Parameters

Name	Description	
id	A Site object's ID	Required

Response Information

Returns

A site's terms and conditions if it exists.

Response Body Format

Terms are return in html formatted string and an empty string if no terms have been set.

On Success:

```
{
  "terms": "<p>When drafting building site terms and conditions, it's
<strong>crucial</strong> to address key aspects like site access, work schedules,
insurance coverage, payment terms, and dispute resolution. These conditions
should clearly outline the responsibilities of all parties involved, including
the builder, subcontractors, and the client, ensuring a smooth and legally sound
building process.</p>"
}
```

On Failure:

Example response if site does not exist:

```
{
  "status": "error",
  "message": "Site not found"
}
```

POST /v1/{clientid}/sites/{id}/terms

Set the terms and conditions for a specific site.

Parameters

Name	Description	
id	A Site object's ID	Required

Request Body format

Include data as a JSON object in the request body. Set "Content-Type" header to "application/json".

Name	Description	
terms	HTML text containing the terms and conditions	
active	Switch site terms on or off	Must be true or false

```
{  
  "terms": "<p>Updated terms and conditions</p>",  
  "active": "true"  
}
```

Response Information

Returns

Success or Failure.

Response Body Format

On Success:

```
{  
  "status": "success",  
  "message": "Site terms have been updated",  
}
```

On Failure:

Example response if site wasn't created:

```
{  
  "status": "error",  
  "message": "Site terms not updated"  
}
```

GET /v1/{clientid}/sites/{id}/qr-code

Generates a QR code for a specific site

Request Information

Parameters

Name	Description	
id	A Site object's ID	Required

Response Information

Returns

QR code if successful otherwise an error message on failure.

On Success:

QR code as an image

On Failure:

Example response if site does not exist:

```
{
  "status": "error",
  "message": "Site not found"
}
```

GET /v1/{clientid}/sites/{id}/logs?fromdate={fromdate}&todate={todate}&status={status}

Get a log of all the check in attempts for a specific site.

Request Information

Parameters

Name	Description	
id	Site ID	Required
fromdate	Date to begin log from	Format: yyyy-mm-dd (defaults to 12 months ago)
todate	Date to stop log at	Format: yyyy-mm-dd (defaults to today)
status	Filter by status of results	pass or fail

Response Information

Returns

Collection of check in attempts with results (if applicable) – returns empty array if no results

Response Body Format

```
[
  {
    "id": "348",
    "result":
    [
      {
        "task-id": "8",
        "task": "Company Policy",
        "state": "valid"
      },
      {
        "task-id": "1",
        "task": "Company Induction ",
        "state": "valid"
      },
      ...
    ],
    "status": "pass",
    "visitorType": "worker",
    "mobile": "+61400000000",
    "userId": "10000",
    "details": {
      "id": "10000",
      "username": "10000",
      "firstname": "Sue",
      "lastname": "Moorehead",
      "mobile": "+61400000000",
      "company": "Company Name"
    },
    "siteId": "1",
```

```
"site": {  
  "id": "1",  
  "site": "Alpha Site"  
},  
"checkedIn": "2024-01-15 06:12:48",  
"checkedOut": null,  
"timestamp": "2024-01-15 06:12:48"  
},  
...  
]
```

GET /v1/{clientid}/kiosks?siteId={siteId}

Get a list of all kiosks for a specific site

Parameters

Name	Description	
siteId	A Site object's ID	Required

Response Information

Returns

A list of kiosks

Response Body Format

```
[
  {
    "id": "afaw3jr3kljafsdklasjf",
    "name": "Reception",
    "siteId": "1",
    "created": "2025-01-31 12:45:10"
  },
  {
    "id": "klj234asdfklj234lkj",
    "name": "Side Gate",
    "siteId": "1",
    "created": "2025-01-31 12:50:01"
  },
  ...
]
```

GET /v1/{clientid}/kioks/{id}

Get a kiosk's details

Request Information

Parameters

Name	Description	
id	A Kiosk object's ID	Required

Response Information

Returns

A kiosk's basic details.

Response Body Format

On Success:

```
{
  "id": "afaw3jr3kljafsdklasjf",
  "name": "Reception",
  "siteId": "1",
  "created": "2025-01-31 12:45:10"
}
```

On Failure:

Example response if kiosk does not exist:

```
{
  "status": "error",
  "message": "Kiosk not found"
}
```

POST /v1/{clientid}/kiosks

Create a new kiosk.

Request Body format

Include data as a JSON object in the request body. Set “Content-Type” header to “application/json”.

Name	Description	
siteId	Site id for the kiosk	Required
name	Kiosk’s name	Required

```
{
  "siteId": "1",
  "name": "Kiosk Name"
}
```

Response Information

Returns

Status, id of the newly added kiosk and link code if successful.

Response Body Format

On Success:

```
{
  "status": "success",
  "message": "Kiosk created",
  "kioskId": "a91dbf94aa1424f41559826f61a33ca1",
  "linkCode": "ORJD"
}
```

On Failure:

Example response if kiosk wasn’t created:

```
{
  "status": "error",
  "message": "Failed to create kiosk"
}
```


PATCH /v1/{clientid}/kiosks/{id}

Update the details for a single kiosk.

Request Information

Parameters

Name	Description	
id	Kiosk object's Id	Required

Request Body Format

Include data as a JSON object in the request body. Set “Content-Type” header to “application/json”. Only includes fields that need to be updated. (Fields that do not need to be updated can be omitted)

Name	Description	
name	Kiosk's name	Required

```
{  
  "name": "South Gate"  
}
```

Response Information

Returns

Success or Failure.

Response Body Format

On Success:

```
{  
  "status": "success"  
  "message": "Kiosk updated."  
}
```

On Failure:

Example response if kiosk wasn't created:

```
{  
  "status": "error",  
  "message": "Failed to update kiosk"  
}
```

GET /v1/{clientid}/kisoks/{id}/link-code

Get a kiosk's code for linking to device.

Request Information

Parameters

Name	Description	
id	A Kiosk object's ID	Required

Response Information

Returns

A code for linking kiosk to a device.

Response Body Format

On Success:

```
{
  "linkCode": "EGUH"
}
```

On Failure:

Example response if kiosk does not exist:

```
{
  "status": "error",
  "message": "Kiosk not found"
}
```

GET /v1/{clientId}/site-visitor-types?siteId={siteId}

Get a list of all visitor types for a specific site

Parameters

Name	Description	
siteId	A Site object's ID	Required

Response Information

Returns

A list of visitor types

Response Body Format

```
[
  {
    "id": "1",
    "name": "Contractor",
    "siteId": "1",
    "requirements": {
      "active": true,
      "programs": ["20", "8"]
    },
    "created": "2025-01-31 12:45:10"
  },
  {
    "id": "2",
    "name": "Truck Driver",
    "siteId": "1",
    "requirements": {
      "active": true,
      "programs": ["5"]
    },
    "created": "2025-01-31 12:50:01"
  },
  ...
]
```

GET /v1/{clientid}/site-visitor-types/{id}

Get a site visitor type details

Request Information

Parameters

Name	Description	
id	A visitor type object's ID	Required

Response Information

Returns

A site visitor type basic details.

Response Body Format

On Success:

```
{
  "id": "1",
  "name": "Contractor",
  "siteId": "1",
  "requirements": {
    "active": true,
    "programs": ["20", "8"]
  },
  "created": "2025-01-31 12:45:10"
}
```

On Failure:

Example response if visitor type does not exist:

```
{
  "status": "error",
  "message": "Site visitor type not found"
}
```

POST /v1/{clientid}/site-visitor-types

Create a new site visitor type.

Request Body format

Include data as a JSON object in the request body. Set “Content-Type” header to “application/json”.

Name	Description	
siteId	Site id for the visitor type	Required
name	Visitor type’s name	Required
requirements	Object with the fields below:	
programs	List of program ids – uses mandatory tasks within program(s) as requirements to be valid when checking in	Required Array of program ids (cannot use guided programs)

```
{
  "siteId": "1",
  "name": "Electrician",
  "requirements": {
    "programs": ["7", "8"]
  }
}
```

Response Information

Returns

Status and id of the newly added visitor type if successful.

Response Body Format

On Success:

```
{
  "status": "success",
  "message": "Site visitor type created",
  "typeId": "20"
}
```

On Failure:

Example response if site visitor type wasn’t created:

```
{
  "status": "error",
  "message": "Failed to create site visitor type"
}
```

PATCH /v1/{clientid}/site-visitor-types/{id}

Update the details for a single site visitor type.

Request Information

Parameters

Name	Description	
id	Site visitor type object's Id	Required

Request Body Format

Include data as a JSON object in the request body. Set "Content-Type" header to "application/json". Only includes fields that need to be updated. (Fields that do not need to be updated can be omitted)

Name	Description	
name	Visitor type's name	
requirements	Object with the fields below:	
programs	List of program ids – uses mandatory tasks within program(s) as requirements to be valid when checking in	Array of program ids

```
{
  "name": "Plumber",
  "requirements": {
    "programs": ["4", "2"]
  }
}
```

Response Information

Returns

Success or Failure.

Response Body Format

On Success:

```
{
  "status": "success"
  "message": "Site visitor type updated."
}
```

On Failure:

Example response if site visitor type wasn't updated:

```
{
  "status": "error",
  "message": "Failed to update site visitor type"
}
```