TrueAbility API V1 Documentation

Base URL: https://app.trueability.com/api/v1
Reset password: https://app.trueability.com/password_resets/new
API access token: https://app.trueability.com/settings

Definitions

AbilityScreen — Performance based test of a user's skills, using software and a live environment. The AbilityScreen defines the assessment requirements and the user instructions.

Assessment — An instance of an AbilityScreen. This consists of 0 or more scenarios, tasks the user must complete, 1 or more nodes (e.g. cloud servers) that comprise the environment.

Assessment Reservation — Assessment build times vary greatly depending on the complexity of the AbilityScreen. In order to guarantee delivery, an assessment is requested via a reservation process.

Reservations must be scheduled a minimum of 24 hours in advance.

Node — A server or resource in the cloud.

Results — Assessments are imaged and graded automatically following completion of the assessment. Imaging can take 15-30 minutes, grading typically takes 1-2 minutes. Scores will be available 30-35 minutes after the completion of the assessment.

Requests

Authentication — the API token is sent with Authorization header, e.g.

```
Authorization: Bearer 00000000
```

Content Type — Requests values are passed as JSON strings using "`application/json`"

Example using curl

```
curl -X POST -H "Content-Type: application/json" \
-H "Authorization: Bearer 00000000" \
-d '{"ability_screen": {"ability_screen_id": "2004", "email": "youruser@example.com"}'
```

Responses

All responses will be formatted using content-type `application/json`

Pagination

Results are pagination into groups of 50 records. Pagination details are included in the json response under the “meta” key.

Example pagination response

```
"meta" : {
    "current_page" : 1,
    "next_page" : 2,
    "prev_page" : null,
    "total_pages" : 8,
    "total_count" : 213
  }
```

Use the query param `page=<:page_number>` to retrieve additional pages.

Status Codes

200 is returned for a successful response
401 is returned when authentication fails
422 is returned when an request is not able to be processed
500 is returned when the api experiences an error.

/ability_screens

An AbilityScreen is an exam that tests a specific set of skills. Each exam has a specific id. This endpoint will give information about all AbilityScreens to which your account has access. You may filter by company by providing a parameter company_id.

```
GET /ability_screens { company_id: <:id> }
GET /ability_screens/<:id>
```

/assessments

Each attempt at an exam is called an assessment. A user that attempts an exam more than once will have one assessment per attempt.

Assessments may be scheduled using the /assessment_reservation endpoint. Assessments scheduled through the /assessments endpoint will be built immediately. Built assessments will expire if not used before MAX_IDLE_TIME. MAX_IDLE_TIME depends on the contracted service agreement.

```
GET /assessments { ability_screen_id: <:ability_screen_id> }
GET /assessments/<:id>
```

Date filtering, for a start or end day, is supported in the index using the format YYYY-MM-DD, e.g.

```
GET /assessments { ability_screen_id:  <:ability_screen_id>,
                   start_at: <:date_window_start>,
                   end_at: <:date_window_end> }
```

/assessment_reservations

A reservation reserves a assessment for some future time. Reservations must be scheduled a minimum of 24 hours in advance. The assessment status should be verified prior to launch. A status of 'notified' indicates the assessment is ready for the candidate. The assessment must be started within the window defined by the AssessmentReservation starts_at and starts_at + MAX_IDLE_TIME.

```
GET /assessment_reservations { ability_screen_id: <:ability_screen_id> }
GET /assessment_reservations/<:id>
POST /assessment_reservations {assessment_reservation: { ability_screen_id:
<:ability_screen_id>, email: <:email_addr>, starts_at: :starting_time_utc, address:
<:address>}}
```

When creating assessments, the address is used to place the user in the nearest operations region. This address defaults to "Austin, TX." Acceptable addresses formats include: Austin, TX; Hong Kong, China; London, England, San Francisco, CA. If the address is blank, we will build the data center in one of our primary US data centers. The data center will be chosen randomly.

/results

Results are normally available approximately thirty (30) minutes after an assessment is completed. The results endpoint provides the score, and relevant assessment information once the assessment has reached the state of archived.

```
GET /results?id[]=<:id1>;id[]=<:id2>;id[]=<:id3>;…id[]=<:idN>
```

Supported query parameters

`id` One or more assessment ids
`uuid` One or more assessment uuids
`assessment_reservation_id` One or more assessment reservation ids
`assessment_reservation_uuid` One or more assessment reservation uuids
`vendor_reservation_id` One or more ids from the vendors (e.g. proctors)
`partner_reservation_id` Oner or more ids from the LMS partner
`company_id` One or more company ids (you must be a member of the company to retrieve results)
`stared_after` Assessments started after this timestamp
`started_before` Assessments started before this timestamp
`graded_after` Assessments graded after this timestamp
`graded_before` Assessments graded before this timestamp

Example of multiple query parameters

```
?company_id[]=1;company_id[]=2;starts_at=2013-10-14T11:19:00
```

```
GET /results/<:id>
```

For convenience, results may also be queried with the Assessment UUID or Assessment Reservation UUID

```
GET /results/assessment_uuid/<:assessment_uuid>
GET /results/assessment_reservation_uuid/<:assessment_reservation_uuid>
```

Internally, the convenience methods redirect to the correct assessment results path, and will return 404 Not found if the assessment does not yet exist.

JSON Object Representation

Singular and Plural examples are given for only for the first example, but are typical of all the endpoints. When you hit an plural action /assessments/ you get the plural response, when you hit a unique endpoint /assessments/12345 you get the singular response.

AbilityScreen

Plural

```
{ "ability_screens": [{"id": "", "name": "", "external_description": "", "company_id":
""}]}
```

Singular

```
{ "ability_screen": {"id": "", "name": "", "external_description": "", "company_id:
""}
```

Assessment

```
{ "assessment": {"id": "",
              "state": "",
              "error": "",
                "url": "",
               "uuid": "",
         "started_at": "",
            "user_id": "",
         "user_email": ""
 }}
```

Company

```
{ "company": {"id": "", "name": ""}}
```

Results

```
{ "results": [{
      "id" : "",                               # assessment id
      "uuid" : "",                             # assessment uuid
      "assessment_reservation_id" : "",
      "assessment_reservation_uuid" : "",
      "user_full_name" : "",                   # name if available, email if not
      "user_email" : "",                       # user’s email address
      "state" : “archived",                    # will always be archived
      "started_at" : "",                       # timestamp when assessment started
      "score" : "",                            # numeric score 0..100
      "score_status" : "pass",                 # <pass|fail>
      "score_threshold" : "",                  # passing threshold 0..100
      "has_results" : true,                    # should always be true
      "duration_in_minutes" : "",              # length of user’s assessment
      "result_url" : "",                       # link to user’s results
      "assessment_files" : []},                # any saved file
       "ability_screen": {
          name: "Certification, Form 1",
          display_name: "Certification",
          id: 1,
          cutoff_score: 0.70
          company: {
            name: “Test, Inc.”,
            id: 1
          }
       }
   ]
}
```

Files associated with Assessments may be compressed using Zlib::Deflate compression. They must be de-compressed before use.

Ruby Example

```
ruby -rzlib -e 'print Zlib::Inflate.new.inflate(STDIN.read)' < filename
```

ServerInstance

```
{"server_instance": {"id": "", "state": ""}}
```

User

```
{"user": {"id": "", "full_name": "", "screen_name": "", "email": "", "gravatar_url": ""}}
```

Examples

GET /ability_screens

```
curl --silent -X GET -H "content-type: application/json" \
 -H "X-API-KEY: AAAAAAAAAAAAAAAAAAA"                     \
 https://app.trueability.com/api/v1/ability_screens
```

```
{
  "ability_screens": [
    {
      "id": 000,
      "name": "Example AbilityScreen",
      "external_description": "<p>Example Environment</p>",
      "company_id": 000
    }
  ]
}
```

GET /ability_screens/0000

```
curl --silent -X GET -H "content-type: application/json" \
 -H "X-API-KEY: AAAAAAAAAAAAAAAAAAA"                      \
 https://app.trueability.com/api/v1/ability_screens/0000
```

```
{
  "ability_screen": {
    "id": 000,
    "name": “Example AbilityScreen",
    "external_description": "<p>Example Environment</p>",
    "company_id": 000
  }
}
```

GET /ability_screen_usage/000

```
curl -X GET -H "Content-Type: application/json" \
-H "X-API-KEY: AAAAAAAAAAAAAAAAAAA"             \
http://app.trueability.com/api/v1/ability_screen_usage/0000
```

```
{
    "ability_screen_usage": {
        "completed": 15,
        "expired": 0,
        "id": 2163,
        "in_progress": 2,
        "total": 24
    }
}
```

GET /assessments?ability_screen_id=0000

```
curl --silent -X GET -H "content-type: application/json" \
 -H "X-API-KEY: AAAAAAAAAAAAAAAAAAA"                      \
 https://app.trueability.com/api/v1/assessments?ability_screen_id=0000
```

```
{
  "assessments": [

  ]
}
```

POST /assessment_reservations

```
curl -X POST -H "content-type: application/json" \
-H "X-API-KEY: AAAAAAAAAAAAAAAAAAAA"              \
-d '{"assessment": {"ability_screen_id": 000, "email": “example@example.com”,
"address": "Hong Kong, China", "starts_at": "2016-04-26T13:33:31.500Z"}}’ \
 https://app.trueability.com/api/v1/assessments
```

```
{
  "assessment_reservation" : {
    "id" : 000,
    "uuid" : "00000000-0000-0000-0000-000000000000",
    "state" : "scheduled",
    "starts_at" : "2016-04-25T22:15:00.000Z",
    "ability_screen_id" : 000,
    "assessment_id" : null,
    "user_id" : 000,
    "created_at" : "2016-04-19T16:31:30.369Z",
    "updated_at" : "2016-04-19T16:31:30.369Z",
    "user_email" : "example@example.com",
    "url" : "http://..."
  }
}
```