

## Tin Can API 0.9 (REST + JSON binding) Client Quick Start

To learn more about the Tin Can API, visit [www.TinCanAPI.com](http://www.TinCanAPI.com)

### Tracking a simple completion

In order to track a completion you will minimally need to know the Tin Can API endpoint of the system you are tracking to, some uniquely identifying information about the person who has completed an activity, and the activity itself.

In most cases, the system you are tracking to will require authentication, so you will need credentials on that system. The simplest way to authenticate is with HTTP basic authentication. OAuth has some advantages, but since it's a little more complex you'll need to go read the authentication section of the TCAPI document if you want to use it.

Consider the case where you have the following information:

Property	Value
Learner e-mail	test.user@example.com
Learner Name	Test User
Activity ID	http://example.com/samples/TinCan101
Activity Title	Example Activity, TinCan 101
Activity Description	Example learning activity about Tin Can.
Endpoint	http://tracking.example.com/TCAPI/
UserName	Testuser
Password	Password

And you want to track that test user has completed activity1.  
You would construct the following message body:

```
{
  "actor" : {
    "mbox" : ["test.user@example.com"],
    "givenName" : ["Test"],
    "familyName" : ["User"]},
  "verb" : "completed",
  "object" : {
    "id" : "http://example.com/samples/TinCan101",
    "type" : "Activity",
    "definition" : {
      "name" : { "en" : "Example Activity, TinCan 101" },
      "description" : { "en" : "Example learning activity
        about Tin Can." }
    }
  }
}
```

And POST it to: “<http://tracking.example.com/TCAPI/statements>”, with the following headers:

## Tin Can API 0.9 (REST + JSON binding) Client Quick Start

To learn more about the Tin Can API, visit [www.TinCanAPI.com](http://www.TinCanAPI.com)

Header	Value
content-type	application/json
authorization	dGVzdHVzZXI6cGFzc3dvcmQ=

Note: the authorization header is generated by BASE64(“username:password”), see [HTTP basic authentication](#) for details. The “en:” in the name and description fields is the language code for English. If giving the name and description in another language, use the [appropriate code](#).

That’s It! Next, we’ll look at tracking a few more details...

---

### Tracking a Score

Suppose the statement you are sending represents a successful completion, with a score of 95%. This can be represented by adding the result section to the message, and the result section would look like this:

```
“result” : {  
    “score” : { “scaled” : .95},  
    “success” : true,  
    “completion” : true  
}
```

Note that this is another way to state “completed”. If you are using the completed verb, you would not have to mention completion here. However, stating completion in the results allows you to use another verb, like “performed” and still report completion. The TinCanAPI document contains the full list of verbs.

---

### Tracking a result of Pass or Fail

Completion doesn’t provide any indication of success – to indicate whether Test User successfully completed Tin Can 101, the verb “passed” or “failed” can be substituted for “completed”. Note that using passed or failed removes the need to specify success or completion in the results section, so going back to the “Tracking a Score” example, this results section could be represented by using the “passed” verb and only including the score, if desired. Mainly the verbs “completed, passed, or failed” are useful as shorthand when the results section is otherwise not needed.

## Tin Can API 0.9 (REST + JSON binding) Client Quick Start

To learn more about the Tin Can API, visit [www.TinCanAPI.com](http://www.TinCanAPI.com)

---

### Tracking a Question

Suppose as part of this learning experience Test User answered the question “Project Tin Can’s focus is: a) content packaging, b) run-time communication, c) metadata, or d) sequencing?”, and Test User answers “b” (which is correct).

This can be reported in the following statement:

```
{
  "actor" : { "mbox" : ["test.user@example.com"]},
  "verb" : "answered",
  "object" : {
    "id" : "http://example.com/samples/TinCan101/Question1",
    "type" : "Activity",
    "definition" : {
      "type" : "question",
      "description" : { "en" : "Project Tin Can's focus
is: a) content packaging, b) run-time communication, c)
metadata, or d) sequencing?" }
    }
  },
  "result" : {
    "response" : "b",
    "success" : "true"
  },
  "context" : {
    "contextActivities" {
      "parent" : {"id" : "http://example.com/samples/
TinCan101" }
    }
  }
}
```

---

### Launch

What if you want to have the tracking system provide launch information, like in SCORM? For web based content, or for any learning activity provider that can respond to an HTTP request, you can still do that. The tracking system will need to be configured to launch your activity.

The tracking system will use a launch link like this:

## Tin Can API 0.9 (REST + JSON binding) Client Quick Start

To learn more about the Tin Can API, visit [www.TinCanAPI.com](http://www.TinCanAPI.com)

```
http://example.scorm.com/TCActivityProvider/  
?endpoint=http://tracking.example.com/TCAPI/  
&auth=OjFjMGY4NTYxNzUwOGI4YWY0NjFkNzU5MWUxMzE1ZGQ1  
&actor={ "name" : ["Test User"], "mbox" :  
["mailto:test.user@example.com"] }  
&registration="760e3480-ba55-4991-94b0-01820dbd23a2"  
&activity_id=http://example.com/samples/TinCan101  
&grouping=http://example.com/samples
```

The above link provides all the information you need to track using the TCAPI (except of course the actual results you want to track).

Note:

- The “auth” parameter is the actual authentication header value to use, not a username or password.
- If the launch link contains any parameters not defined here, they must simply be passed back in any statements.
- If the launch link contains a “grouping” or “registration” parameter, it must be added to the context, as shown below.
- A single content/activity endpoint to be launched can track multiple activities. The specific activity to be launched will be specified in the “activity\_id” parameter.
- An actual link would have URI Parameter encoded parameters, and no line breaks.

So if you wanted to report that the specified person passed the specified activity, that statement would look like this:

```
{  
  "actor" : {"mbox" : ["test.user@example.com"] },  
  "verb" : "passed",  
  "object" : "id" : "http://example.com/samples/TinCan101",  
  "context" : {  
    "registration" : "760e3480-ba55-4991-94b0-01820dbd23a2",  
    "contextActivities" : {  
      "grouping" : { "id" : "http://example.com/samples"}  
    }  
  }  
}
```

Note that only identifying information about the actor and object are passed. It is not necessary (but it is allowed) to pass detailed information about actors or objects if the tracking system has or will get those details from another source. In this case, since the tracking system provided the details about these objects in the launch link, we know it doesn't need the details repeated.

## Tin Can API 0.9 (REST + JSON binding) Client Quick Start

To learn more about the Tin Can API, visit [www.TinCanAPI.com](http://www.TinCanAPI.com)

---

### Bookmarking

Particularly for activities that are launched by an LMS, or other tracking system, and may not have a way to persist their own data, it is important to be able to save the learner's place in their experience of that activity. Suppose the learner is on "lesson 2" within TinCan101. The value to be stored is simply "lesson 2", and it can be stored by issuing an HTTP PUT to the following URL, with the contents "lesson 2":

```
http://tracking.example.com/TCAPI/activities/state/?activityId=http://  
/example.com/samples/TinCan101/760e3480-ba55-4991-94b0-  
01820dbd23a2&actor={"mbox" : ["test.user@example.com"]}  
&stateId=cmi.location
```

This is making use of the state portion of the TinCan API for our specific example, the general form is:

```
http://tracking.example.com/TCAPI/activities/?activity  
Id=id&actor=actorObject[&registrationId=id]&stateId=id
```

Note:

- Activity ID and agent, shown in ***bold italic*** will need to be URI Parameter encoded, and have not been here for readability.
  - Any state key can be used to store any document
  - Retrieving state values is a matter of issuing a HTTP GET to the same URL
  - Registration is required if provided, otherwise it is left off.
  - The authentication header should be set as when posting a statement, and the content header depending on the content being stored.
- 

### More Information

This document only covers some frequently used capabilities of the Tin Can API. Please refer to the complete documentation [here](#) for more information.