HTTP Endpoints

SEETechnology

**Contents**

[1 Public pages 2](#_Toc394313485)

[1.1 The Home page 2](#_Toc394313486)

[1.2 The About us page 2](#_Toc394313487)

[1.3 The Find a partner page 2](#_Toc394313488)

[1.4 The Search page 2](#_Toc394313489)

[1.5 The Organization profile page 3](#_Toc394313490)

[1.5.1 Skills and Endorsements module 3](#_Toc394313491)

[1.5.2 Recommendations module 4](#_Toc394313492)

[1.5.3 Members module 4](#_Toc394313493)

[1.5.4 Organizations module 4](#_Toc394313494)

[1.5.5 Sidebar 4](#_Toc394313495)

[2 Personal pages 5](#_Toc394313496)

[2.1 My Account page 5](#_Toc394313497)

[2.2 My Organizations page 5](#_Toc394313498)

[3 Edit organization page 5](#_Toc394313499)

[3.1 General section 6](#_Toc394313500)

[3.2 Description section 6](#_Toc394313501)

[3.3 Head section 7](#_Toc394313502)

[3.4 Administrators section 7](#_Toc394313503)

[3.5 Skills and expertise section 7](#_Toc394313504)

[3.6 Members section 8](#_Toc394313505)

[3.7 Recommendations section 8](#_Toc394313506)

[3.8 Organizations section 9](#_Toc394313507)

[3.9 Reports section 9](#_Toc394313508)

[4 System administration pages 10](#_Toc394313509)

[4.1 The System Administration page 10](#_Toc394313510)

[4.2 The Top-level organizations page 10](#_Toc394313511)

[4.3 The Error log page 10](#_Toc394313512)

[4.4 The Activity log page 11](#_Toc394313513)

[4.5 The Edit the About us page 11](#_Toc394313514)

[4.6 The Edit the Newsetters section on the Home page page 11](#_Toc394313515)

# Public pages

## The Home page

Request: GET /

Response: the full HTML markup of the page, except the Tweets module which is generated by the Twitter widget on the client side.

## The About us page

When the page is loaded:

Request: GET /About

Response: the full HTML markup of the page.

When the user clicks the Send button to submit the Contact form:

Request: POST /About

Response: the full HTML markup of the page with a notification bar on the top which shows that the form is successfully submitted.

## The Find a partner page

When the page is loaded:

Request: GET /Find

Response: the full HTML markup of the page, except the Google Maps which is loaded by client side JavaScript code.

When the user clicks the Filter button:

Ajax request: POST /Find?q={filter expression}

Ajax response: JSON which contains the organizations matching the filter criteria. Only those properties are returned which are required to display the organizations on the map with a pin and a tooltip.

When the user clicks a pin on the map:

Ajax request: GET /Find?id={organization ID}

Ajax response: JSON which contains the details of the selected organization.

## The Search page

When the page is loaded:

Request: GET /Search?k={search keyword}

Response: the full HTML markup of the page including the search results.

When the user clicks the Search button:

Request: GET /Search?k={search keyword}&a={A|N|D|L|S}&c={country code}

This is a GET request to ensure that the user can bookmark the results.

The “a” parameter contains the selected search area (see the Find in dropdown list):

* “A” or missing or invalid: Anywhere
* “N”: Organization name
* “D”: Organization description
* “L”: Organization location
* “S”: Organization skill or expertise

Response: the full HTML markup of the page including the search results.

## The Organization profile page

When the page is loaded:

Request: GET /org/{organization slug}

Response: the full HTML markup of the page.

### Skills and Endorsements module

When the user clicks the “…” button for a skill:

Ajax request GET /org/{organization slug}/{skill slug}

Ajax response: JSON which contains the endorsers.

When the user clicks the Endorse button at the bottom of the module:

Ajax request: GET /org/{organization slug}/TopSkills

Ajax response: JSON which contains 5 skills to endorse.

When the user types a new skill into the modal dialog:

Ajax request: GET /Skills/{keyword}

Ajax response: JSON which contains the top 5 matching skills.

When the user clicks the Endorse button in the modal dialog:

Ajax request: POST /org/{organization slug}/Endorse

The request body contains the list of skills.

Ajax response: JSON which contains the result of the operation.

### Recommendations module

When the user clicks the Like button:

Ajax request: POST /org/{organization slug}/Like

Ajax response: JSON which contains the result of the operation.

When the user clicks the Unlike button:

Ajax request: POST /org/{organization slug}/Unlike

Ajax response: JSON which contains the result of the operation.

When the user clicks the Send button in the modal dialog:

Ajax request: POST /org/{organization slug}/Recommend

The request body contains the text of the recommendation.

Ajax response: JSON which contains the result of the operation.

### Members module

When the user clicks the “…” button:

Ajax request GET /org/{organization slug}/Members

Ajax response: JSON which contains the members.

When the user clicks the Yes button in the modal dialog:

Ajax request POST /org/{organization slug}/Join

Ajax response: JSON which contains the result of the operation.

### Organizations module

When the user clicks the Send button in the modal dialog:

Ajax request POST /org/{organization slug}/CreateSubOrg

The request body contains the name of the organization to create.

Ajax response: JSON which contains the result of the operation.

### Sidebar

When the user clicks the Report button on the sidebar:

Ajax request: POST /org/{organization slug}/Report

The request body contains the details of the report.

Ajax response: JSON which contains the result of the operation.

# Personal pages

## My Account page

When the page is loaded:

Request: GET /MyAccount

Response: the full HTML markup of the page.

When the user clicks the Delete and then the Yes, I understood, delete my account buttons:

Request: DELETE /MyAccount

Response: redirect to the /MyAccountDeleted URL. The response must clear the authentication cookie.

When the user clicks the Update from LinkedIn button:

Request: POST /MyAccount

The client side LinkedIn API is used to query the details of the current user from LinkedIn, and then it is posted to the server with this POST request. The body of the request contains the user’s properties.

Response: the full HTML markup of the page (full page reload).

When the user clicks the Save button:

Request: POST /MyAccountEmail

Response: the full HTML markup of the page (full page reload)

## My Organizations page

When the page is loaded:

Request: GET /Manage

Response: the full HTML markup of the page.

When the user clicks the Leave and its confirmation button:

Request: POST /Manage

The request body contains the ID of the organization to leave.

Response: the full HTML markup of the page.

# Edit organization page

When the page is loaded:

Request: GET /Manage/{organization slug}

Response: The HTML markup of the page with closed sections.

## General section

When the user clicks the Save button in the Edit name dialog:

Ajax request: PUT /Manage/{organization slug}/Name

The request body contains the new name and abbreviation of the organization.

Ajax response: JSON which contains the result of the operation.

When the user clicks the Change parent link:

Ajax request: GET /Manage/{organization slug}/Parents

Ajax response: JSON which contains the possible parent organizations.

When the user clicks the Save button in the Change parent dialog:

Ajax request: PUT /Manage/{organization slug}/Parent

The request body contains the ID of the new parent organization.

Ajax response: JSON which contains the result of the operation.

When the user clicks the Upload button in the Change logo dialog:

Request: PUT /Manage/{organization slug}/Logo

The request body contains the new logo file.

Response: the full HTML markup of the page.

When the user clicks the Save button in the Edit contact dialog:

Ajax request: PUT /Manage/{organization slug}/Contact

The request body contains the new contact details of the organization.

Ajax response: JSON which contains the result of the operation.

When the user clicks the Save button in the Edit URLs dialog:

Ajax request: PUT /Manage/{organization slug}/URLs

The request body contains the new URLs of the organization.

Ajax response: JSON which contains the result of the operation.

## Description section

When the user clicks the Description header to open the section:

Ajax request: GET /Manage/{organization slug}/Description

Ajax response: the HTML markup of the description.

When the user clicks the Save button in the Edit description diaog:

Ajax request: PUT /Manage/{organization slug}/Description

The request body contains the new description text.

Ajax response: JSON which contains the result of the operation.

## Head section

When the user clicks the Head header to open the section:

Ajax request: GET /Manage/{organization slug}/Head

Ajax response: JSON which contains the details of the organization head.

When the user clicks the Save button in the Change organization head dialog:

Ajax request: PUT /Manage/{organization slug}/Head

The request body contains the details of the new head.

Ajax response: JSON which contains the result of the operation and the details of the new head.

## Administrators section

When the user clicks the Administrators header to open the section:

Ajax request: GET /Manage/{organization slug}/Administrators

Ajax response: JSON which contains the details of the organization admins.

When the user clicks the Yes button in the Remove an organization administrator dialog:

Ajax request: DELETE /Manage/{organization slug}/Administrators

The request body contains the ID of the admin to remove.

Ajax response: JSON which contains the result of the operation.

When the user clicks the Save button in the Add new organization administrator dialog:

Ajax request: POST /Manage/{organization slug}/Administrators

The request body contains the details of the new admin.

Ajax response: JSON which contains the result of the operation and the details of the newly added admin.

## Skills and expertise section

When the user clicks the Skills and expertise header to open the section:

Ajax request: GET /Manage/{organization slug}/Skills

Ajax response: JSON which contains the list of the organization’s skills.

When the user clicks the Remove button in the Remove a skill or expertise dialog:

Ajax request: DELETE /Manage/{organization slug}/Skills

The request body contains the ID of the skill to remove.

Ajax response: JSON which contains the result of the operation.

When the user clicks the Save button in the Add new skill or expertise dialog:

Ajax request: POST /Manage/{organization slug}/Skills

The request body contains the name of the new skill to add.

Ajax response: JSON which contains the result of the operation and the details of the newly added skill.

## Members section

When the user clicks the Members header to open the section:

Ajax request: GET /Manage/{organization slug}/Members

Ajax response: JSON which contains the list of the organization’s members and the people who want to join to the organization.

When the user clicks the Remove button in the Remove member dialog:

Ajax request: DELETE /Manage/{organization slug}/Members

The request body contains the ID of the member to remove.

Ajax response: JSON which contains the result of the operation.

When the user clicks the Add button in the Add new member dialog:

Ajax request: POST /Manage/{organization slug}/Members

The request body contains the details of the new member to add.

Ajax response: JSON which contains the result of the operation.

When the user clicks the Yes, approve request or the No, reject request button in the Join request dialog:

Ajax request: PUT /Manage/{organization slug}/Members

The request body contains the ID of the member to change and the operation (approve or reject).

Ajax response: JSON which contains the result of the operation.

## Recommendations section

When the user clicks the Recommendations header to open the section:

Ajax request: GET /Manage/{organization slug}/Recommendations

Ajax response: JSON which contains the details of all recommendations.

When the user clicks the Show or Hide button for a recommendation:

Ajax request: PUT /Manage/{organization slug}/Recommendations

The request body contains the ID of the recommendation to change and the operation (show or hide).

Ajax response: JSON which contains the result of the operation.

## Organizations section

When the user clicks the Organizations header to open the section:

Ajax request: GET /Manage/{organization slug}/SubOrgs

Ajax response: JSON which contains the details of the sub-organizations.

When the user clicks the Delete button in the Delete sub-organization dialog:

Ajax request: DELETE /Manage/{organization slug}/SubOrgs

The request body contains the ID of the sub-organization to delete.

Ajax response: JSON which contains the result of the operation.

When the user clicks the Add button in the Add new sub-organization dialog:

Ajax request: POST /Manage/{organization slug}/SubOrgs

The request body contains the name and the abbreviation of the new sub-organization.

Ajax response: JSON which contains the result of the operation.

When the user clicks the Approve button in the Approve sub-organization request dialog:

Ajax request: PUT /Manage/{organization slug}/SubOrgs

The request body contains the ID, the name, the abbreviation and the value of the Assign administrative permissions to the requestor checkbox, and the operation (approve).

Ajax response: JSON which contains the result of the operation.

When the user clicks the Reject button in the Reject sub-organization request dialog:

Ajax request: PUT /Manage/{organization slug}/SubOrgs

The request body contains the ID of the request and the operation (reject).

Ajax response: JSON which contains the result of the operation.

## Reports section

When the user clicks the Reports header to open the section:

Ajax request: GET /Manage/{organization slug}/Reports

Ajax response: JSON which contains the list of all reports.

When the user clicks the Delete button in the Deleting a report dialog:

Ajax request: DELETE /Manage/{organization slug}/Reports

The request body contains the ID of the report to delete.

Ajax response: JSON which contains the result of the operation.

# System administration pages

## The System Administration page

When the page is loaded:

Request: GET /Admin

Response: the full HTML markup of the page.

When the user clicks one of the Export… buttons:

Request GET /Admin/Export/Organizations or GET /Admin/Export/Users

Response: JSON which contains the absolute URL of the export file on the server.

## The Top-level organizations page

When the page is loaded:

Request: GET /Admin/TopOrgs

Response: the full HTML markup of the page.

When the user clicks the Delete button:

Ajax request: DELETE /Admin/TopOrgs/{ID}

Ajax Response: JSON which contains the result of the operation.

When the user enters a name into the Add new top-level organization dialog:

Ajax request: GET /Admin/TopOrgs?Name={name}

Ajax response: JSON which contains whether an organization exists with the given name.

When the user enters an abbreviation into the Add new top-level organization dialog:

Ajax request: GET /Admin/TopOrgs?Abbreviation={name}

Ajax response: JSON which contains whether an organization exists with the given abbreviation.

When the user clicks the Add button:

Ajax request: PUT /Admin/TopOrgs

The request body contains the name and the abbreviation of the new organization.

Ajax response: JSON which contains the result of the operation.

## The Error log page

When the page is loaded:

Request: GET /Admin/ErrorLog

Response: the full HTML markup of the page.

When the user clicks one error log entry:

Ajax request: GET /Admin/ErrorLog/{ID}

Ajax response: JSON which contains the details of the error log entry.

## The Activity log page

When the page is loaded:

Request: GET /Admin/ActivityLog

Response: the full HTML markup of the page.

## The Edit the About us page

When the page is loaded:

Request: GET /Admin/AboutUs

Response: the full HTML markup of the page.

When the user clicks the Save button:

Ajax request: PUT /Admin/AboutUs

The request body contains the new content of the About Us page.

Ajax response: JSON which contains the result of the operation.

## The Edit the Newsetters section on the Home page page

When the page is loaded:

Request: GET /Admin/Newsletters

Response: the full HTML markup of the page.

When the user clicks the Save button:

Ajax request: PUT /Admin/Newsletters

The request body contains the new content of the Newsletters section on the Home page.

Ajax response: JSON which contains the result of the operation.