HTTP Endpoints

SEETechnology

# 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

TODO

# 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/EditAboutUs

Response: the full HTML markup of the page.

When the user clicks the Save button:

Ajax request: POST /Admin/EditAboutUs

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/EditNewsletters

Response: the full HTML markup of the page.

When the user clicks the Save button:

Ajax request: POST /Admin/EditNewsletters

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.