

StrandMaps Advanced Tutorial

Strand Maps Service Advanced Tutorial



Unknown macro: 'float'

This tutorial shows how to use the Strand Map Service and UCARConnect Search Service to build a custom map browser to display specific maps and display relevant UCARConnect resources in the Information Bubble for the benchmarks. The tutorial builds off of knowledge from the [Basic Tutorial](#) and covers more advanced topics such as creating a custom strand selector to provide access to a subset of maps. Since the advanced tutorial will involve modifying the user interface created for the [Basic Tutorial](#). You can [download the files for the basic tutorial](#) and start from there. |

Scenario/Goal: Create an advanced Strand Maps user interface with a custom strand selector that shows only high school level earth science resources from the *DLESE* and *Harvard-Smithsonian Digital Video Collections*. [View a Demo](#)

Requirements

To proceed with this tutorial, you will need:

- Familiarity with the [Science Literacy Maps](#) user interface
- Familiarity with the [SMS JavaScript API documentation](#)
- Familiarity with the topics discussed in the [Basic Tutorial](#)
- Web space to host your new user interface.

Although not required, it is useful to have the following:

- A set of resources to be aligned to the AAAS benchmarks
- A collection of resources in UCARConnect

Step 1.a: Determine which maps to provide access to

To create a custom strand selector we need to know the Strandmap Service IDs, which are referred to in this tutorial as SMS IDs, for the maps or part of maps that we intend to display. To acquire a map ID, navigate to the specific map in the [Science Literacy Maps](#) and copy the ID from the browser's address bar. For example, the Plate Tectonics map, located at <http://strandmaps.ndsl.org/?id=SMS-MAP-0049> would have the ID SMS-MAP-0049.

By using the Science Literacy maps user interface and navigating the astronomy related maps, we know that the map IDs that we want are

- SMS-GRD-2063 for Use of Earth's Resources
- SMS-MAP-0049 for Plate Tectonics

The Strandmap Service API provides the ability to limit the maps displayed to any level within the maps. It is possible to only show specific strands or specific grade bands of maps. To determine these sub IDs, you'll need to use the [JSON explorer](#) and traverse the JSON returned.

Step 1.b: Determining map IDs

Since we only want to provide access to the high school level (grades 9-12) of the astronomy maps, we'll need to use the JSON explorer to determine which IDs correspond to these subsections of the astronomy maps. The strand maps data model is based on a hierarchy that shows the relationships between benchmarks, the relationship between a benchmark to a grade band and strand, the relationship between a strand or grade level to a map, and so on. Therefore, to determine which ID for a grade band to use, we'll need to start with a benchmark that is within that grade band for a particular map.

For example, to determine the ID of the grades 9-12 grade band in the Plate Tectonics map:

- Navigate to the [Plate Tectonics map](#)
- Click on a benchmark from the 9-12 grade band to open the information bubble, such as *The theory of plate tectonics provides an explanation for a diverse array of seemingly unrelated phenomena...*
- Copy the SMS ID for that benchmark, which is displayed in the benchmark full text. The ID of the example benchmark is **SMS-BMK-0034**.
- Navigate to the [JSON explorer](#)
- Paste the SMS ID in the ObjectID field and click the View JSON button.
- The JSON data is returned with the service response.

To make traversal easier, the strand and grade levels are listed above the gray JSON code area. Grade levels IDs are expressed by *SMS-GRD* and strand IDs are expressed by *SMS-STD*. For our example, only one grade level is listed so we can use the ID returned in our strand selector (SMS-GRD-0047). Some benchmarks may belong to multiple strands or grades, especially if the benchmark is found in multiple maps. In this case, you should click the *explore* link next to the possible IDs to verify that the grade band ID belongs to the appropriate map.

Repeat this process to get the 9-12 grade band ID for the Use of Earth's Resources map (SMS-GRD-2063).

Step 2: Create your custom strand selector

Starting with [the files for the basic tutorial](#) we will create a custom strand selector.

The `<strandSelector>` div tag is used to automatically populate the user interface with the default strand selector. Since we are replacing the default strand selector with a custom selector, remove that div tag entirely and replace with the following selector code.

```

<div id="smsNavigation">
  <form action="" method="get" id="smsBrowse">
    <select name="id" id="id">
      <option value="">-- Select a Topic --</option>
      <option value="SMS-GRD-2063">Use of Earth's Resources</option>
      <option value="SMS-GRD-0047">Plate Tectonics</option>
    </select>
    <input type="submit" value="Go" id="submitBrowse" />
  </form>
</div>

```

This snippet generates a new selector widget that sets the ID parameter on the page when submitted. The ID parameter is used by the Strandmap Service to pull up a particular map.

Step 3: Create a tab for Educational Resources

To display resources, we need to create a tab for them in the InfoBubble. The InfoBubble is a singleton object that controls the content, appearance, features and behaviors associated with the information bubble widget that appears in the maps. It allows you to add or remove tabs, register actions that are performed when the bubble is opened and closed, and access data associated with the bubble. The methods available for the infoBubble object are explained in the [API documentation](#).

The Strand Map Service provides two resource tabs that automatically perform searches over the UCARConnect collections. The *Aligned Tab* only returns resources that have been explicitly aligned to the current benchmark. The *Related Tab* returns resources that contain similar keywords as the current benchmark as defined by the service.

Within the *setUpStrandMap* function, which was described in [Step 4.a. of the basic tutorial](#), add the following code snippet to include a tab for aligned resources. *(Code snippets below will change?)*

```
infoBubble.addBuiltinTab("nsdlaligned", "Top Picks");
```

If you want a tab for related resources, include the following code snippet after the aligned resources tab

```
infoBubble.addBuiltinTab("nsdlrelated", "Related Resources");
```

The tabs are displayed in the order they are added, so add this snippet before adding the NSES Standards tab if you want the resources to appear first. As a reminder, the *setUpStrandMap* function is a user-defined callback function that is used to provide additional instructions to your interface.

Step 4. Options: Search specific collections

You can select specific UCARConnect collections and resources to search over for the *Aligned Resources* tab, the *Related Resources* tab or both. To do so you must include an array with the collection keys in your javascript file prior to initiating the service. Add the following before your *setUpStrandMap* function:

```
var smsCollectionsAligned = new Array('dlese.org', '843818');
```

This will limit the Aligned Resources tab to only the DLESE and Harvard-Smithsonian Digital Video collections. If you want to limit the Related Resources tab, use the following snippet.

```
var smsCollectionsRelated = new Array('dlese.org', '843818');
```

By default, both tabs will search over the all UCARConnect repository resources if no collection IDs are provided. Note: Not all benchmarks have aligned or related resources. If no resources are found, a message indicating so will appear in the tab's content.

To find the collection IDs, go to the [NSDL.org Search User Interface](#) and search over a keyword of your choosing. Collections that provide metadata for each resource are listed on the right side of the page. Hover your mouse pointer over the collection you want to choose and click the *About this Collection* link. On the next page, look at the browser URL and copy the value for the collection ID, which is the value provided after the *'collection/'* directory. For example if the collection URL is <http://nsdl.org/search/collection/843818/>, then the value you should include in your collections array is 843818. The collection ID value should be added to your array to only search over resources in that collection. You can include any number of collections to your array.