

---

# **go-jsbox-location Documentation**

***Release 0.2.0***

**Praekelt Foundation**

June 02, 2015



<b>1</b>	<b>LocationState Class</b>	<b>3</b>
<b>2</b>	<b>GoogleMaps</b>	<b>5</b>
<b>3</b>	<b>OpenStreetMap</b>	<b>9</b>
<b>4</b>	<b>Providers</b>	<b>13</b>
<b>5</b>	<b>Testing utilities</b>	<b>15</b>
<b>6</b>	<b>Indices and tables</b>	<b>17</b>



Contents:



---

## LocationState Class

---

**class LocationState** (*name, opts*)

A state which requests a location from the user, and sets the user data through getting the location data from the Google Maps API. It may also request a second prompt from the user to further refine the location if the first prompt wasn't clear enough.

### Arguments

- **name** (*string*) – name used to identify and refer to the state
- **opts.question** (*string\_or\_LazyText*) – The question to first display to the user. Defaults to What is your address?.
- **opts.map\_provider** (an instance of `:class:location.providers.utils.Provider`) – The provider to use when searching for locations. Defaults to an instance of `:class:location.providers.googlemaps.GoogleMaps`.
- **opts.refine\_question** (*string\_or\_LazyText*) – The question to display to the user when selecting a location from a list if the first search query wasn't clear enough. Defaults to Please select your location from the following:
- **opts.error\_question** (*string\_or\_LazyText*) – The question to display to the user when no locations are found for their search term. It will keep requesting until results are found. Defaults to Error: No results for your search term. Please try another search term.
- **opts.continue\_session** (*boolean*) – whether or not this is the last state in a session. Defaults to `true`.
- **opts.send\_reply** (*boolean*) – whether or not a reply should be sent to the user's message. Defaults to `true`.
- **opts.next** (*function\_or\_string\_or\_object*) – The state that the user should visit after this state. May either be the name of the next state, an options object representing the next state, or a function of the form `f (content)` returning either, where `content` is the input given by the user. If `next` is `null` or not defined, the state machine will be left in the current state. See `State.set_next_state()`. Defaults to `null`
- **opts.options\_per\_page** (*integer*) – The maximum limit for the amount of choices on each page. Defaults to 8.
- **opts.characters\_per\_page** (*integer*) – The maximum limit for the amount of characters on each page. Defaults to 160. Whichever one of `characters_per_page` or `option_per_page` is reached first will be chosen.

- **opts.next\_text** (*string*) – The text to display for the next page option. Defaults to Next.
- **opts.previous\_text** (*string*) – The text to display for the previous page option. Defaults to Previous.
- **opts.namespace** (*string*) – The namespace to use when storing the contact details, ie. `location:.....` Defaults to `location`.
- **opts.events** (*object*) – Optional event name-listener mappings to bind.

Example:

```
self.states.add('states:example-locationState', function(name) {
  return new LocationState(name, {
    question: ["Welcome to the location app.",
              "What is your current address?"].join("\n"),
    next: "states:end",
    previous_text: "Prev",
  });
});
```

Each provider has its own `.fixture(...)` function for conveniently creating HTTP resource fixtures for its location queries. See the documentation for each provider on how to use these.



---

## GoogleMaps

---

**class GoogleMaps** (*opts*)

An address search provider for Google Maps.

### Arguments

- **opts.api\_url** (*string*) – The URL of the Google Maps geocode API. Defaults to `'http://maps.googleapis.com/maps/api/geocode/json'`.
- **opts.extract\_address\_label** (*function*) – Function that extracts a human-friendly label from a GoogleMaps API result. See `GoogleMaps.extract_address_label()` for the function signature and default implementation.
- **opts.extract\_address\_data** (*function*) – Function that extracts data to store on a contact from a GoogleMaps API result. See `GoogleMaps.extract_address_data()` for the function signature and default implementation.

**static extract\_address** (*result*)

Takes a result from the Google Maps API and returns a `AddressResult()` for it.

### Arguments

- **result** (*object*) – A raw GoogleMaps result.

**Returns** An `Addressresult()`.

Calls `GoogleMaps.extract_address_label()` and `GoogleMaps.extract_address_data()` for the label and data respectively.

**static extract\_address\_data** (*result*)

Extracts address data to store on a contact when an address is selected. See `LocationState.store_contact_data()` for a description of how this data is stored.

Returns the object:

```
{
  formatted_address: result.formatted_address
}
```

May be overridden using the `extract_address_data` class option.

### Arguments

- **result** (*object*) – A location result from the Google Maps API.

**Returns object** An object of key-value pairs to store in contact data.

**static** `extract_address_label (result)`

Returns the value of `result.formatted_address` as a human-friendly display name for a Google Maps location result.

May be overridden using the `extract_address_label` class option.

**Arguments**

- **result** (*object*) – A location result from the Google Maps API.

**Return string** The label to display for this result.

**static** `init ()`

Initialization function invoked during state initialization.

**Arguments**

- **im** (*InteractionMachine*) – The state's `InteractionMachine()` instance.

**static** `search (query_text)`

Return an ordered list of locations matching the query via a promise that is fulfilled when the search results are ready.

Returns an empty list if an error occurs while accessing the GoogleMaps API.

**Arguments**

- **query\_text** (*string*) – The search query.

**Returns** A promise that yields the list of search results.

**fixture (opts)**

Returns an HTTP resource fixture for a GoogleMaps location query.

**Arguments**

- **opts.query** (*string*) – The address that is to be queried. Required.
- **opts.address\_list** (*array of strings*) – A list of `formatted_address`'s that should be sent in the response. If `response_data` is included, this will be ignored. Defaults to `''[]`.
- **opts.request\_url** (*string*) – URL for the HTTP request. Defaults to `"http://maps.googleapis.com/maps/api/geocode/json"`.
- **opts.response\_data** (*object*) – An that object that represents the response from the Google Maps API. This overrides the response data generated from `opts.address_list`. Example response data:

```
{
  status: "OK",
  results: [
    {
      formatted_address: "1 Baker Street",
    },
    {
      formatted_address: "2 Baker Street",
    },
  ],
}
```

Usage:

```
tester
    .setup(function(api) {
        api.http.fixtures.add(
            GoogleMaps.fixture({
                request: "New Street",
                address_list: [
                    "New Street 1", "New Street 2",
                ],
            });
        });
    });
```



---

## OpenStreetMap

---

**class** `OpenStreetMap` (*opts*)

An address search provider for Open Street Map.

### Arguments

- **opts.api\_url** (*string*) – The URL of the Open Street Map API. Defaults to `'http://open.mapquestapi.com/nominatim/v1/search.php'`.
- **opts.extract\_address\_label** (*function*) – Function that extracts a human-friendly label from an Open Street Map API result. See `OpenStreetMap.extract_address_label()` for the function signature and default implementation.
- **opts.extract\_address\_data** (*function*) – Function that extracts data to store on a contact from an Open Street Map API result. See `OpenStreetMap.extract_address_data()` for the function signature and default implementation.
- **hard\_boundary** (*boolean*) – Whether to limit results to a fixed bounding box. Defaults to *true*.
- **address\_limit** (*integer*) – Maximum number of search results to return. Defaults to 30.
- **bounding\_box** (*array*) – Bounding box to limit search results to. Should be provided in the form `[left_edge, top_edge, right_edge, bottom_edge]`, e.g. `["-18.3273", "-33.7652", "18.937", "-34.3329"]` for South Africa. Defaults to `["-180.0", "90.0", "180.0", "-90.0"]` for worldwide search.

See <http://open.mapquestapi.com/nominatim/> for a complete description of the Open Street Map search API.

**static** `extract_address` (*result*)

Takes a result from the OpenStreetMap API and returns a `AddressResult()` for it.

### Arguments

- **result** (*object*) – A raw OpenStreetMap result.

**Returns** An `Addressresult()`.

Calls `OpenStreetMap.extract_address_label()` and `OpenStreetMap.extract_address_data()` for the label and data respectively.

**static** `extract_address_data` (*result*)

Extracts address data to store on a contact when an address is selected. See `LocationState.store_contact_data()` for a description of how this data is stored.

Returns the object:

```
{
  formatted_address: result.display_name
}
```

May be overridden using the `extract_address_data` class option.

#### Arguments

- **result** (*object*) – A location result from the Open Street Map API.

**Returns object** An object of key-value pairs to store in contact data.

**static extract\_address\_label** (*result*)

Returns the value of `result.display_name` as a human-friendly display name for an Open Street Map location result.

May be overridden using the `extract_address_label` class option.

#### Arguments

- **result** (*object*) – A location result from the Open Street Map API.

**Return string** The label to display for this result.

**static init** ()

Initialization function invoked during state initialization.

#### Arguments

- **im** (*InteractionMachine*) – The state's `InteractionMachine()` instance.

**static search** (*query\_text*)

Return an ordered list of locations matching the query via a promise that is fulfilled when the search results are ready.

Returns an empty list if an error occurs while accessing the Open Street Map API.

#### Arguments

- **query\_text** (*string*) – The search query.

**Returns** A promise that yields the list of search results.

**fixture** (*opts*)

Returns an HTTP resource fixture for an OpenStreetMap location query.

#### Arguments

- **opts.query** (*string*) – The address that is to be queried. Required.
- **opts.address\_list** (*array of strings*) – A list of `display_name`'s that should be sent in the response. If `response_data` is included, this will be ignored. Defaults to ```[]`.
- **hard\_boundary** (*boolean*) – Whether the request should limit results to a fixed bounding box. Defaults to `true`.
- **address\_limit** (*integer*) – The maximum number of search results that should be requested. Defaults to 30.
- **bounding\_box** (*array*) – The bounding box to submit in the search request. Should be provided in the form `[left_edge, top_edge, right_edge, bottom_edge]`, e.g. `["-18.3273", "-33.7652", "18.937", "-34.3329"]` for South Africa. Defaults to `["-180.0", "90.0", "180.0", "-90.0"]` for worldwide search.

- **opts.request\_url** (*string*) – URL for the HTTP request. Defaults to "http://open.mapquestapi.com/nominatim/v1/search.php".
- **opts.response\_data** (*object*) – An object that represents the response from the OpenStreetMap API. This overrides the response data generated from `opts.address_list`.  
Example response data:

```
[
  {
    display_name: "1 Baker Street",
  },
  {
    display_name: "2 Baker Street",
  },
]
```

Usage:

```
tester
  .setup(function(api) {
    api.http.fixtures.add(
      OpenStreetMap.fixture({
        request: "New Street",
        address_list: [
          "New Street 1", "New Street 2",
        ],
      });
    });
  });
```





---

## Providers

---

Utilities and base classes to use when writing providers.

**class `AddressResult()`**

A result returned by `:meth:Provider.search`.

### Arguments

- **label** (*string*) – A human-friendly label for the search result.
- **data** (*object*) – A map of key-value pairs to store on a person's contact object if they select this result. See `LocationState.store_contact_data()` for a description of how this data is stored.

**class `FixtureParameterMissingError()`**

Error raised when a parameter required to create a fixture was not provided.

### Arguments

- **message** (*string*) – An explanation of which parameter was not provided.

**class `Provider()`**

A base class for address search providers.

Extensions to this class should implement `init()` and `search()`.

**static `init()`**

Initialization function invoked during state initialization.

### Arguments

- **im** (*InteractionMachine*) – The state's `InteractionMachine()` instance.

**Returns** May return a promise that fires once initialization is complete.

**static `search(query_text)`**

Return an ordered list of locations matching the query via a promise that is fulfilled when the search results are ready.

### Arguments

- **query\_text** (*string*) – The search query.

**Returns** A promise that yields the list of `AddressResult()` instances.

**class `ProviderNotImplementedError()`**

Error raised when a method on a provider has not been implemented.

### Arguments

- **message** (*string*) – An explanation of which method was not implemented.

---

## Testing utilities

---

**assert\_address\_result** (*result*, *label*, *data*)

**Arguments**

- **result** (*object*) – The result object assert on.
- **label** (*string*) – The expected label.
- **data** (*object*) – The expected data. Defaults to {formatted\_address: *label*}.



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



## A

`AddressResult()` (class), [13](#)

`assert_address_result()` (built-in function), [15](#)

## F

`fixture()` (built-in function), [6](#), [10](#)

`FixtureParameterMissingError()` (class), [13](#)

## G

`GoogleMaps()` (class), [5](#)

## L

`LocationState()` (class), [3](#)

## O

`OpenStreetMap()` (class), [9](#)

## P

`Provider()` (class), [13](#)

`ProviderNotImplementedError()` (class), [13](#)