

Browser

- Requirements
- Invocation
- Open a page internally
- Open a page externally
- Configure Browsers
- Get the currently configured browsers
- Example

Requirements

The examples on this page use the **Prototype** library, which is included by default inside a portal. However, the same concepts may be applied using other implementations.

Invocation

This command is executed immediately in a synchronous way.

Open a page internally

When a page needs to be opened internally within Studio, use this JavaScript code to dispatch the appropriate action:

```
dispatch($H({
  controller: 'portal.browser',
  action: "internalOpen",
  args: ["http://www.appcelerator.com"].toJSON()
}).toJSON());
```

Open a page externally

When a page needs to be opened externally in the OS default browser, use this JavaScript code to dispatch the appropriate action:

```
dispatch($H({
  controller: 'portal.browser',
  action: "externalOpen",
  args: ["http://www.appcelerator.com"].toJSON()
}).toJSON());
```

Configure Browsers

Configure the Studio's browsers by detecting the installed browsers and adding them to the Eclipse Browsers preference page.

The call returns a Map for the newly added browsers. The map contains the mapping from a browser location to a browser name.

In case no browser was added, an empty map is returned.

```
var addedBrowsers = dispatch($H({
  controller: 'portal.browser',
  action: "configureBrowsers"
}).toJSON());
```

Get the currently configured browsers

Returns the currently configured browsers in the Eclipse Browsers preference page.

The call returns a Map for the detected browsers. The map contains the mapping from a browser location to a browser name.

"null" location in the result indicates the Eclipse default browser.

```
var browsers = dispatch($H({
  controller: 'portal.browser',
  action: "getConfiguredBrowsers"
})).toJSON();
```

Example

You may place that dispatch code above inside a link listener, just like that:

```
myLink = $('my-link');
myLink.observe('click', function(e) {
  if (typeof(dispatch) !== 'undefined') {
    dispatch($H({
      controller: 'portal.browser',
      action: "externalOpen",
      args: ["http://www.appcelerator.com"].toJSON()
    })).toJSON();
  }
  return false;
});
```