

Titanium User Information

- Requirements
- Description
- Invocation
- Get the User Information
- Sample of use

Requirements

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

Description

These set of actions can retrieve a Titanium User log-in information from the Studio.

Invocation

This command is executed immediately in a synchronous way.

Get the User Information

The following *dispatch* call will get the Titanium user basic information:

```
userInfo = dispatch($H({
  controller: 'portal.user',
  action: "getUser"
}).toJSON()).evalJSON();
```

The returned *user information* JSON holds this information:

key	description
id	A user ID
name	The user-name
email	The user-email
hash	An email MD5 hash that can be used to retrieve the user's Gravatar

Sample of use

From [studio3-sdk](#) repository (*user.js*)

```

/**
 * Render the items that will display the Titanium-User information
 */
render : function() {
  userDiv = $('user');
  with(Elements.Builder) {
    if( typeof (console) !== 'undefined' && typeof (dispatch) !== 'undefined') {
      console.log("Dispatching the 'getUser' action on the 'portal.user'
controller...");
      userInfo = dispatch($H({
        controller: 'portal.user',
        action: "getUser"
      })).toJSON().evalJSON();
      userTable = table({
        "border": "1",
        "style": "border-collapse:collapse"
      }, tbody(
        tr(
          th("ID"),
          th("Name"),
          th("Email"),
          th("Image")),
        tr(
          td(userInfo["id"]),
          td(userInfo["name"]),
          td(userInfo["email"]),
          td(img({"src": "http://www.gravatar.com/avatar/" + userInfo["hash"]})))
        )
      ));
      userDiv.appendChild(userTable);
    }
  }
}
}

```