Custom Datapane API
Version 1.0, Rev A

1. Background

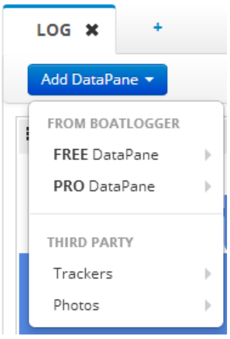
The BoatLogger site is built up by datapanes. These datapanes can be added or removed from the user's site and they can be configured to show various content.

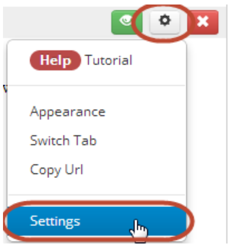
The normal datapanes are defined and hosted by BoatLogger. Through the Custom Datapane API it is however possible to host the content of a datapane at another server and display the datapane as if it appears in the BoatLogger site. This enables other developers to easily interface with the BoatLogger eco system.

2. Technical Background

The custom datapane is shown in an iFrame in the user's site. The URL and some other key information about the custom datapane are defined in the BoatLogger site by an administrator.

The custom datapane then appears in the Add datapane menu under the THIRD PARTY heading in an appropriate category. This makes it available for all users to add on their site.





The developer actually specifies two URL's. One is for the datapane content and the other URL is for the settings that is opened when the user presses the datapane settings button as the picture to the left.

The Tutorial is currently not available for custom datapanes.

To make it possible to make richer datapanes some parameters are send with the URL to the custom datapanes. These parameters provide information about the site visited and also provides an API key to be used to exchange information with the site (see more details below).

3. Configuration

Before a datapane can be made available to the general users it needs to be defined by a BoatLogger administrator. Thereafter it appears in the THIRD PARTY menu in the 'Add DataPane' menu.

The information required by BoatLogger is:

Datapane name	This is the name appearing in the add datapane menu and as default title of the data pane
Description	A description of what this datapane does
Provider	The name of the person or company providing the content
Private Token	This is given by BoatLogger and uniquely defines this datapane.
Applicable sites	What type of BoatLogger sites that the datapane make sense for (Yacht sites, race sites, flotilla sites etc)
Category	What category the datapane shall appear under in the 'Add DataPane' menu (ultimately decided by BoatLogger).

Content height — The default content height for this datapane (can be changed by the user in the settings for the datapane). An example could be “600”.

Content Url — The URL for the content. An example could be : “http://www.thehostingsite.com/ myDatapaneContent.html” (the ending html is not necessary and the full path is up to the developer)

Settings Url — The URL for the settings. An example could be : “http://www.thehostingsite.com/ myDatapaneSettings.html”

4. The datapane

4.1. The URL

BoatLogger calls the content or settings Url with a Url like:

http://www.thehostingsite.com/myDatapaneContent.html?isOwner=true&userKey=1&userSiteKey=1&token=01234569

The parameters are as follows:

isOwner — A parameter specifying whether current user is the site owner or not. If true, settings and more detailed information can be displayed.

userKey — Each user in the BoatLogger site has a unique number called userKey. This can identify users and might be useful for the API in the future.

userSiteKey — Each site that is created has a unique number. A user may have one yacht site and multiple race or flotilla sites. Each site will have a unique userSiteKey.

Token — This is only used if the datapane want to use the Web Service API to access the site data (see more in 4.2 below).
This is the Public Token which will usually change for each call. It is only valid one hour after the call. This public token is similar to, but different from, the BLWS key used in the normal Web Service API. The public token is used together with the private token (received during setup from BoatLogger) to perform authorization as described below.

4.2. Using the Web Service API

There is no requirement that the datapane accesses the Web Service API, but it can be useful to exchange information with the specific BoatLogger site such as logbook information, weather information, boat facts, pictures etc.

To access the Web Service API, authorization is necessary. The authorization mechanism from a datapane works in the same way as from a normal client except that from normal clients, we use **BLWS** as the authorization scheme. To authorize custom datapane access, we use **BLDP** as the authorization scheme. The authorization thus has the following form:

Authorization: BLDP {PublicKey}:{PrivateKey}

Public Key is obtained every time the datapane displayed to the user, and Private Key is a key generated by the administrator at BoatLogger when the new custom datapane is added.

There is a number of ways to use the Web Service API. It is recommended to access from the server side, and not the client side due to the risk of exposing the Private Key to snoopers. However, there is no restriction in Web Service API calls using Javascript. The snippet below show how easily one can query photos using the API.

```
$.ajax({
    url: "http://boatlogger.com/api/Picture",
    type: "GET",
    headers: {
        "Authorization": "BLDP " + publicToken + ":" + privateToken
    },
}).success(function (data) {
    alert("API Call success");
}).fail(function (msg) {
    alert("API call failed");
});
```

For more information about Web Service API call, please refer to existing API documentation at http://www.boatlogger.com/Feature/Downloads .

4.3. A custom datapane example

The following code is the complete content of a sample custom datapane. To use it, save it e.g. as a filename called "datapane.html" and place it on a webserver, then specify this as the datapane URL to use with BoatLogger:

```
<html>
<head>
   <title>Custom datapane</title>
   <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
</head>
<body>
   <h3 style="margin: 0px; padding: 0px">Sample DataPane</h3>
   <table>
      <tr>
         <td>Token</td>
         <td><span id="tokenSpan"></span></td>
      </tr>
      <tr>
         <td>User Site Key</td>
         <td><span id="UserSiteKey"></span></td>
      </tr>
      <tr>
         <td>Command</td>
         <td><input id="tbCommand" type="text" /><input id="btnSendCommand" type="button" 
value="button" /></td>
      </tr>
   </table>
</body>

<script type="text/javascript">
   function getParameterByName(name) {
      name = name.replace(/[\[]/, "\\[").replace(/[\]]/, "\\]");
      var regex = new RegExp("[\\?&]" + name + "=([^&#]*)"),
         results = regex.exec(location.search);
      return results == null ? "" : decodeURIComponent(results[1]);
   }

   $(function () {
      var isOwner = getParameterByName("isOwner");
      var userKey = getParameterByName("userKey");
      var userSiteKey = getParameterByName("userSiteKey");
      var publicToken = getParameterByName("token");
      var privateToken = "bcda0defd34a48a8b46fb01239448025"
      $("#tokenSpan").html(publicToken);
      $("#UserSiteKey").html(userSiteKey);
      $("#btnSendCommand").click(function () {
         alert("Clicked it! The value was " + $("#tbCommand").val() );
      });
   });

</script>
</html>
```