



HCL Domino Volt

Using JavaScript

Presented by Christopher Dawes
Leap/Volt Consultant, SME, Developer

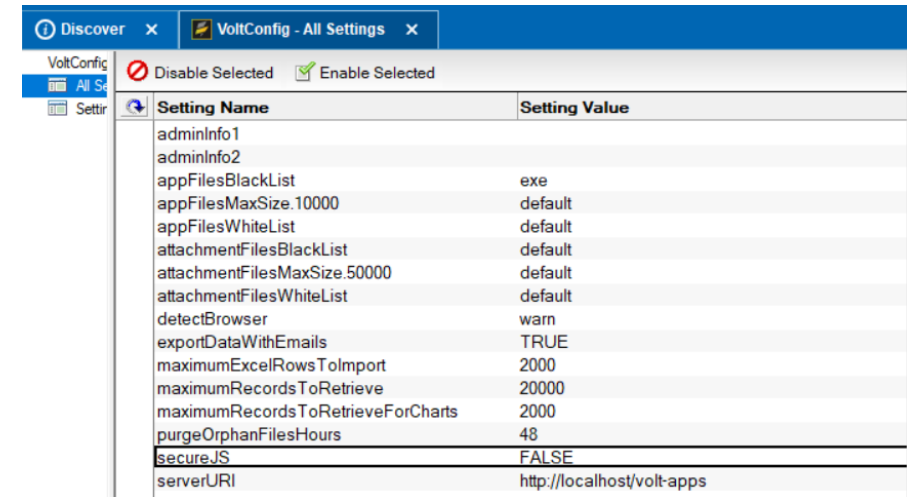


Agenda

- JavaScript Sandbox
- Model (Interface vs Data)
- JavaScript Events
- Global Variables
- Global Functions
- Debugging
- Error Handling
- Triggering Services
- Working with Tables
- Working with Lists
- Validation Before Submit
- Additional References

JavaScript Sandbox

- By default Volt javaScript development is contained within a sandbox, this means you **do not** have access to many standard javaScript objects and functions
- Can modify in the Volt Config, **secureJS**
- If in sandbox:
 - Use `get()`, `set()` instead of `[n]` notation
 - Use `===`, `!==` instead of `==`, `!=`
 - Can't use keyword **this**
 - May limit use of third-party javaScript libraries
- Apps must be re-deployed for secureJS change to take effect
- Refer to https://leap.hcldoc.com/help/topic/LEAPv9/LEAP/ref_jsapi_javascript_security.html



The screenshot shows the 'VoltConfig - All Settings' window. It features a table with two columns: 'Setting Name' and 'Setting Value'. The table lists various settings such as 'adminInfo1', 'appFilesBlackList', 'appFilesMaxSize.10000', 'appFilesWhiteList', 'attachmentFilesBlackList', 'attachmentFilesMaxSize.50000', 'attachmentFilesWhiteList', 'detectBrowser', 'exportDataWithEmails', 'maximumExcelRowsToImport', 'maximumRecordsToRetrieve', 'maximumRecordsToRetrieveForCharts', 'purgeOrphanFilesHours', 'secureJS', and 'serverURI'. The 'secureJS' setting is highlighted in blue and has a value of 'FALSE'. The 'serverURI' setting has a value of 'http://localhost/volt-apps'. Above the table, there are buttons for 'Disable Selected' and 'Enable Selected'.

Setting Name	Setting Value
adminInfo1	
adminInfo2	
appFilesBlackList	exe
appFilesMaxSize.10000	default
appFilesWhiteList	default
attachmentFilesBlackList	default
attachmentFilesMaxSize.50000	default
attachmentFilesWhiteList	default
detectBrowser	warn
exportDataWithEmails	TRUE
maximumExcelRowsToImport	2000
maximumRecordsToRetrieve	20000
maximumRecordsToRetrieveForCharts	2000
purgeOrphanFilesHours	48
secureJS	FALSE
serverURI	http://localhost/volt-apps

Volt JavaScript Model

- There are 2 layers: Interface vs Data
- The Interface pertains to anything visual and the data to the data
- Interface variables: app, form, page, item
- Data variables: BO, BOA
- Variables are scoped
- Interact with items by their ID, an Advanced property

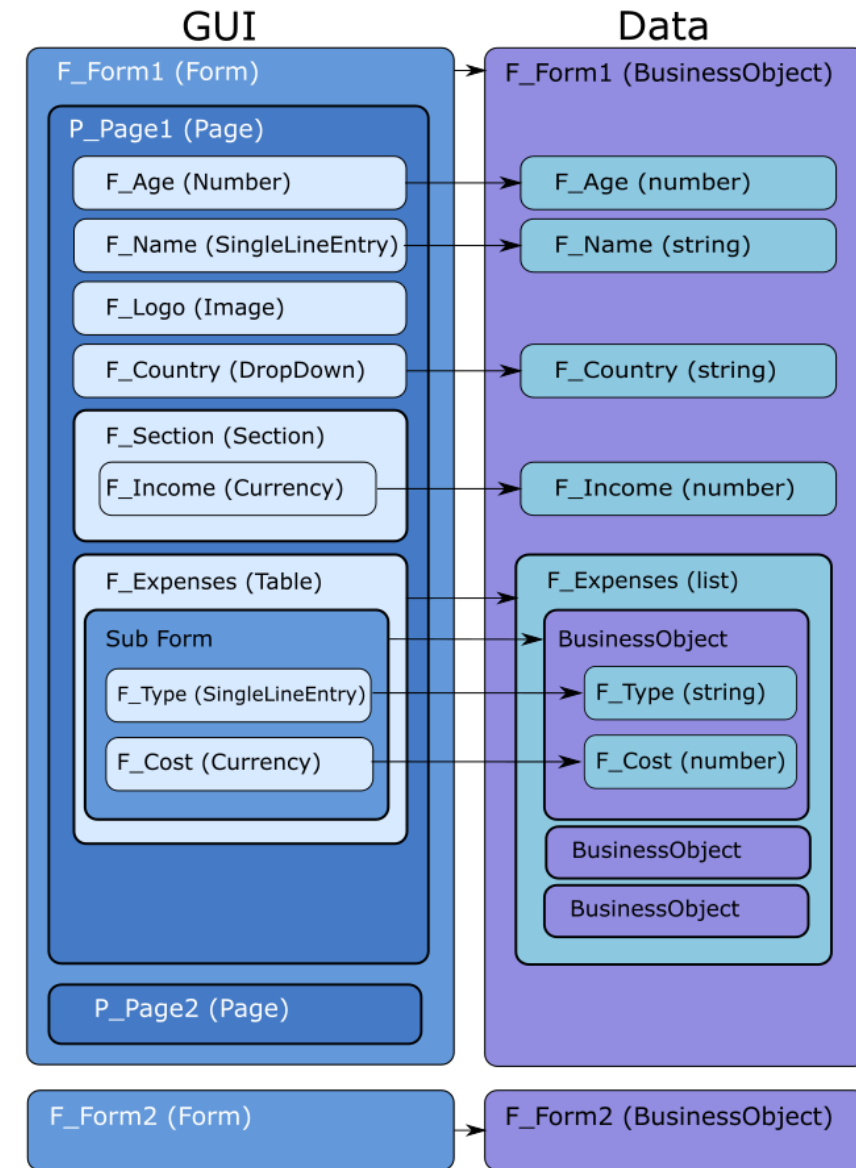
Edit Single Line Entry Properties

Basic **Advanced** Formula Events

ID: ⓘ
F_SingleLine1

Data Label: ⓘ

- **Set Meaningful IDs!!**



Interface Examples

- Common Interface Functions

```
app.getCurrentUser()  
app.getUID()  
app.getUrlParameters() / app.getUrlParameter(parm)
```

```
form.getBO()  
form.getCurrentPage()  
form.getPage(thePageId)  
form.selectPage(thePageId)
```

```
item.getValue()  
item.getVisible()  
item.getId()  
item.setFocus()  
Item.setContent(c)
```

There are many more...

https://leap.hcldoc.com/help/topic/LEAPv9/LEAP/ref_jsapi_user_interface_objects.html

Data Examples

- Common Data Functions

BO.getCurrentStage()

BO.<itemId>.getValue()

BO.<itemId>.setValue()

BO.<itemId>.setValid(valid, msg)

BO.<itemId>.setRequired(r)

BOA.isValid()

BOA.setRequired(r)

BOA.isValid()

There are many more...

https://leap.hcldoc.com/help/topic/LEAPv9/LEAP/ref_jsapi_ref_data_objects.html

JavaScript Events

- Every object in Volt has events that can be used to trigger custom JavaScript
- **Application:** onStart
- **Form:** onNew, onLoad, validateButtonPressed, ...
- **Page:** onShow, onNavigateAway, ...
- **Item:** onClick, onItemChange, onShow, ...

The screenshot shows the HCL Domino Volt Beta interface. The top navigation bar includes 'Forms', 'Access', 'Stages', 'Style', 'Events' (selected), 'Settings', and 'Validation'. The main content area is titled 'Form 1 - F_Form1 » validateButtonPressed'. It contains a description: 'This event is invoked when a stage action button is pressed. The parameter "pActionId" is the ID of the st'. Below this, there are sections for 'Predefined Actions' (Run a Formula, Call a Service) and 'Custom Actions'. A code editor shows the following JavaScript code:

```
//can't submit if client hours is less than 50
if(pActionId === "S_Submit") {
  var sum = BO.F_Number1.getValue() + BO.F_Number2.getValue() + BO.F_Nu
  if(sum < 50) {
    alert("Project hours do not meet the minimum requirement (50).");
    return false;
  }
}
```

The image shows three overlapping screenshots of the 'Events' tab in different dialog boxes:

- Edit Form Properties:** Lists events such as afterSave, beforeSave, onDataReceived, onDestroy, onHide, onLoad, onNew, onShow, onShowActionButtons, and validateButtonPressed.
- Edit Page Properties:** Lists events such as onHide, onNavigateAway, onRemoveFromNavigation, onRestoreToNavigation, and onShow.
- Edit Single Line Entry Properties:** Lists events such as onClick, onHide, onInvalid, onItemBlur, onItemChange, onItemFocus, onItemLiveChange, onMouseOut, onMouseOver, onShow, and onValid.

Global Variables

- Volt provides a global mechanism for registering functions and variables that can be accessed from anywhere within your application.

```
app.getSharedData()...
```

Variable Definition and use:

```
app.getSharedData().myvar = 5;
```

```
If(BO.F_Number.getValue() < app.getSharedData().myVar) {
```

```
    ...
```

```
}
```


Global Functions

- Function Definition

```
app.getSharedData().myfunction = function(num1, num2) {  
    return num1 + num2;  
}
```

- Function Use

```
var res = app.getSharedData().myfunction(2, 3);  
var res = app.getSharedData().myfunction(BO.F_Number1.getValue(),  
BO.F_Number2.getValue());  
F_Number3.setValue(res);
```

Debugging

- Use “debugger;” in code to set a breakpoint
- Open browser debug tools (F12, Ctrl+Shift+i)
- Trigger code, then step through (each browser uses different keys):

```
257 var s = ""; s = ""
258 for(var i=0;i<tbl.getLength();i++) { i = 0, tbl = {__bol: {...}}
259 debugger;
260 var row = tbl.get(i);
261 var rColor = row.F_SingleLine1.getValue();
262 if(s !== "")
263     s += ",";
264
265     s += rColor;
266 }
267 BO.F_Paragraphtext4.setValue(s);
268
```

- Can use console.log(...) if secureJS=false
- Demonstration

onClick

This event is invoked whenever the user clicks on the item

Predefined Actions:

- Run a Formula
- Call a Service ⓘ

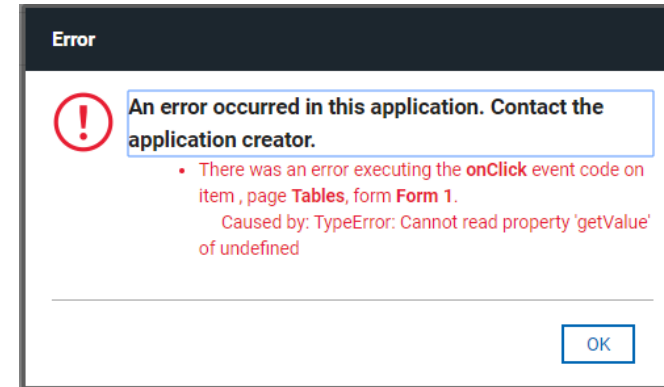
Custom Actions:

```
var tbl = BO.F_Table1;
var s = "";
for(var i=0;i<tbl.getLength();i++) {
debugger;
    var row = tbl.get(i);
    var rColor = row.F_SingleLine1.getValue();
    if(s !== "")
        s += ",";

    s += rColor;
}
BO.F_Paragraphtext4.setValue(s);
```

Error Handling

- Volt has built-in error handling



- You may want to trap all errors and handle them differently (sometimes good for debugging tricky problems as well)

```
try {  
  ...  
} catch(err) {  
  alert(err);  
}
```

Triggering Services

- Services can be triggered using JavaScript. This is helpful if you need to setup the inputs of the service programmatically before the service executes.

```
form.getServiceConfiguration('SC_theService').callService();
```

- To execute code after a service finishes, use an onCallFinished handler

```
var srv = form.getServiceConfiguration('SC_theService');  
var hdl = srv.connectEvent("onCallFinished", function(success)  
{  
  if(success) {  
    //put code in here to execute when service is done  
  }  
});
```

- Depending on where you connect the event you may also need to disconnect it.
form.disconnectEvent(hdl);

Services in a Loop

- Services are asynchronous
- If the services are not related, could use a for loop (but this is rarely true)
- Need to do it differently, this will call one after another until complete

```
app.getSharedData().serviceCount = 0;  
form.getServiceConfiguration("SC_GetWorkshopCount").callService();
```

```
var srv = form.getServiceConfiguration("SC_Service");  
srv.connectEvent("onCallFinished", function(success) {  
  if(success) {  
    ...  
  
    if(app.getSharedData().serviceCount <= 5) {  
      form.getServiceConfiguration("SC_Service").callService();  
    }  
    app.getSharedData().serviceCount++;  
  }  
});
```

- Example

<https://hclwiki.atlassian.net/wiki/spaces/HL/pages/461610/Working+with+Services+-+calling+Services+in+a+Loop>

Triggering Submit

- You may want to create your own custom button, rather than using the ones provided.
 - Can place it wherever you want
 - More styling options
 - Can use images

- Form Submit buttons are accessed in a special way.

```
var actionButtons = form.getStageActions();
for(var i=0; i<actionButtons.length; i++){
    if(get(actionButtons, i).getId() === "btnID")
        get(actionButtons, i).activate();
}
```

Working With Tables

- Get column value from selected row in a table

```
var sel = item.getSelection();
var color = sel.F_SingleLine.getValue();
```
- Create Row in table

```
var newRow = theTable.createNew();
newRow.F_SingleLine.setValue("Red");
theTable.add(newRow);
```
- Iterate over table items

```
var tbl = BO.F_Table;
for(var i=0;i<tbl.getLength();i++) {
  var row = tbl.get(i);
  var rColor = row.F_SingleLine.getValue();
  ...
}
```
- Demonstration

Working with Lists

- Multi-Select List values are a concatenated string: Red__#__Yellow__#__Blue
- List Options are an object {title: "The Display Value", value: "The Saved Value"}

```
var nOpts = new Array();  
nOpts.push({title:"Red", value:"Red"});  
...  
page.F_Dropdown.setOptions(nOpts);  
page.F_SelectOne.setOptions(nOpts);  
page.F_SelectMany.setOptions(nOpts);
```

- **beforeOptionsUpdate** event
- Demonstration

Validation Before Submit

- You may have a need for custom validation before a form can be submitted, for this we use the **validateButtonPressed** event.
- Detect which submit button was triggered, can return **false** to cancel the submit action.

```
//can't submit if client hours is less than 50
if(pActionId === "S_Submit") {
  var sum = BO.F_Number1.getValue() + BO.F_Number2.getValue() + BO.F_Number3.getValue();
  if(sum < 50) {
    alert("Project hours do not meet the minimum requirement (50).");
    return false;
  }
}
```

- Demonstration

Additional Resources

- For additional detail on the javaScript API refer to https://help.hcltechsw.com/domino_volt/1.0/ref_javascript_api.html
- Domino Volt Wiki (<https://hclwiki.atlassian.net/wiki/spaces/HDV/overview>)
- HCL Leap Wiki (<https://hclwiki.atlassian.net/wiki/spaces/HL/pages/462131/Introduction+to+HCL+Leap>)