
NF
23
CONNECTING
THE WORLD

Intro to Niagara JS

JJ Frankovich

Software Engineer @ Tridium

TRIDIUM

Overview

- What is Niagara JS
  - Object Model - bajaScript
  - Widget Model - bajaux
  - Utilities - many live in js-ux.jar module.
- Go over an existing Java technology (refresher) vs JS

Why learn Niagara JS?

- You should already know Niagara Java APIs
- Browser and Mobile Support
- Differentiate your product
- Bajaux Widgets work in Browsers and Workbench

Data Model (no ui yet)

- Subclassing BComponent.java, the bread and butter of Niagara

```
@NiagaraProperty(
  name = "message",
  type = "String",
  defaultValue = "Hello Niagara Forum"
)
@NiagaraType
class BEchoComponent extends BComponent
  public String toString(Context context){
    return getMessage();
  }
}
```

Data Model (Bajascript)

- Subclassing Component.js, the bread and butter of Niagara

```
class EchoComponent extends Component
    toString(cx) {
      return this.get('message');
    }
}
```

BComponent.java set and add

```
    BEchoComponent c = new BEchoComponent();
    c.set("message", BString.make("message from Java"));

    c.add("newMessage", BString.make("another message from Java"));
  }
```

Component.js set and add

```
const c = baja.$('slides:EchoComponent');
c.set({
  slot: 'message',
  value: 'switch to Bajascript'
});
c.add({
  slot: 'newMessage',
  value: 'it even runs in the browser!'
});

print(c);
```

Run

Component.js set and add

```
const c = baja.$('slides:EchoComponent');
c.set({
  slot: 'message',
  value: 'switch to Bajascript'
});
c.add({
  slot: 'newMessage',
  value: 'it even runs in the browser!'
});

print(c);
```

```
hi
switch to Bajascript, it even runs in the browser!
```

Type Extensions - Connecting JS to Java

```
@NiagaraType(agent = @AgentOn(types = "slides:EchoComponent"))
@NiagaraSingleton
public final class BEchoComponentTypeExt
  extends BBajaScriptTypeExt
  implements BIOffline  //Components->Yes, History->No.
{
  @Override
  public JsInfo getTypeExtJs(Context cx) { return JS_INFO; }

  private static final JsInfo JS_INFO = JsInfo.make(
    BOrd.make("module://slides/rc/baja/EchoComponent.js");
}
```

Component.js Tips

- Component.js will handle all core Niagara stuff like Properties, Actions, and Topics
- TypeExtensions good for custom toString and functions.
- Think about functions that might be used as a BFormat in UxMedia

```
class ControlPoint extends Component
  getValueWithFacets(cx) { //@since Niagara 4.13
    return this.getStatusValue().getValue().toString();
  }
  //%valueWithFacets%
```

Data Model (Simples)

- Subclassing BSimple.java, the data of Niagara

```
class BDemoSize extends BSimple {
    public BDemoSize(double width, double height) {
      this.width = width;
      this.height = height;
    }

    public String encodeToString() {
       return width + ";" + height;
    }
    ...
}
```

Data Model (JS)

- Subclassing Simple.js, the data of Niagara

```
class DemoSize extends baja.Simple {
    constructor(width, height) {
      super();
      this.$width = width; //$ for private class variable
      this.$height = height;
    }

    encodeToString() {
      return this.$width + ';' +  this.$height;
    }
    ...
}
```

Simple.js Tips

- DefaultSimple.js automatically provided for any Simples without a custom implementation
- Server side toString recorded
- Custom Simples can help with when you want to work with unmounted instance (newCopy)
- Ensure encodeToString and decodeFromString matches Java.

BOrd.java

- Find me my data!
- BAJA's Object Resolution Descriptor

```
BControlPoint point =
  BOrd.make("station:|slot:/forum23/Temperature").get();
System.out.println("point: " + point);
```

Ord.js

- Ord.js, find my data in a Browser!

```
return baja.Ord.make('station:|slot:/forum23/Temperature').get()
  .then((point)=>{
      window.console.log('point: ' + point);
  });
```

Run

Ord.js

- Ord.js, find my data in a Browser!

```
return baja.Ord.make('station:|slot:/forum23/Temperature').get()
  .then((point)=>{
      window.console.log('point: ' + point);
  });
```

Run

```
point: 16.3 °C {ok}
```

NF 23

Subscriber.java

- Find me the changes in my data.

```
    BControlPoint point =
BOrd.make("station:|slot:/forum23/Temperature").get();
    Subscriber subscriber = new Subscriber() {
        @Override
        public void event(BComponentEvent event) {
          System.out.println("point:" + point);
        }
    }
    subscriber.subscribe(point);
```

Subscriber.js

- Find me the changes in my data in the browser.

```
const subscriber = new baja.Subscriber();
baja.Ord.make('station:|slot:/forum23/Temperature').get({ subscriber })
  .then((point)=>{
      subscriber.attach("changed", function (prop, cx) {
        window.console.log(point);
      });

      return asyncUtils.waitInterval(6000)
        .then(()=> subscriber.unsubscribeAll());
  })
```

Run

Subscriber.js

- Find me the changes in my data in the browser.

```
const subscriber = new baja.Subscriber();
baja.Ord.make('station:|slot:/forum23/Temperature').get({ subscriber })
  .then((point)=>{
      subscriber.attach("changed", function (prop, cx) {
        window.console.log(point);
      });

      return asyncUtils.waitInterval(6000)
        .
  })
```

```
24.5 °C {ok}
23.1 °C {ok}
```

Subscriber.js Tips

- subscriber.unsubscribeAll()
- 4.13+, Compare subscriber spy pages
  spy:/niagaraNetwork/workbenchConnections
  spy:/nav/localhost/station/Services/BoxService/ssession

Remote Station | niagaraNetwork | workbenchConnections | Session

Connection	
address	127.0.0.1:62125
user	admin
connectTime	30-Mar-23 3:47 PM EDT
app	Workbench 4.13.0.145

Subscriptions (4 Components)
/Services/HistoryService/historyGroupings
/Services/JobService
/forum23/Temperature
/forum23/Temperature/proxyExt

Remote Station | nav | localhost | station | Services | BoxService | ssession | ServerSession1 | station$243a

Subscriptions (2 Components)
/forum23/Temperature
/forum23/Temperature/proxyExt

ASync Java

- Multiple threads
- requires synchronization
- take as long as you want (Just not on Engine Thread)

```
new Thread() {
    public void run() {
        Thread.sleep(1000);
        System.out.println("1 second later Java");
    }
}.start();
```

Async JS

- Single threaded
- no synchronization
- keep the browser rendering
- Niagara JS uses Promises

```
return asyncUtils.waitInterval(1000)
  .then(() => {
      window.console.log("1 second later js");
  })
```

Run

Async JS

- Single threaded
- no synchronization
- keep the browser rendering
- Niagara JS uses Promises

```
return asyncUtils.waitInterval(1000)
  .then(() => {
      windo
  })
```

```
1 second later js
```

What is a Promise?

- A unit of asynchronous work
- Can succeed or fail
- **`try/catch/finally`** semantics
- Can be grouped together

Promise example

```
return Promise.all([
  baja.Ord.make('station:|slot:/Services').get(),
  baja.Ord.make('station:|slot:/Drivers').get()
])
.then(([services, drivers]) => {
  console.log('resolved both components');
  console.log(services.getNavOrd());
  console.log(drivers.getNavOrd());
});
```

Run

Promise example

```
return Promise.all([
  baja.Ord.make('station:|slot:/Services').get(),
  baja.Ord.make('station:|slot:/Drivers').get()
])
.then(([services, drivers]) => {
  console.log('resolved both components');
  console.log(services.getNavOrd());
  console.log(drivers.getNavOrd());
});
```

```
resolved both components
local:|station:|slot:/Services
local:|station:|slot:/Drivers
```

try/catch/finally

```
return Promise.try(function () {
  return baja.Ord.make('station:|slot:/Services2').get();
})
.catch(function (err) {
  console.log('failed to resolve Services2: ' + err);
})
.finally(function () {
  console.log('this runs no matter what');
});
```

Run

try/catch/finally

```
return Promise.try(function () {
  return baja.Ord.make('station:|slot:/Services2').get();
})
.catch(function (err) {
  console.log('failed to resolve Services2: ' + err);
})
.finally(function () {
  console.log('this runs no matter what');
});
```

```
BoxError: Slot doesn't exist: Services2 at local:|sta
BoxError: Slot doesn't exist: Services2 at local:|sta
failed to resolve Services2: BoxError: Slot doesn't e
this runs no matter what
```

Promise.js Tips

- Return the Promise or add a catch handler
- add JSDoc for your methods
- IntelliJ ESLint support warnings

```
/**
 * @returns {Promise}
 */
function runDemoProperly() {
  return baja.Ord.make("station:|slot:/Temperature").get();
}

function runDemo() {
  baja.Ord.make("station:|slot:/Temperature").get();
}
```

Promise returned from get is ignored

Add '.then()' Alt+Shift+Enter More actions...

Java Imports

- Imports are basic, but important

```
import javax.baja.sys.BComponent;
import javax.baja.sys.Context;
import javax.baja.sys.Property;
import javax.baja.sys.Sys;
import javax.baja.sys.Type;

import javax.baja.alarm.BAlarmClass;
import javax.baja.alarm.BAlarmDatabase;
```

Imports in JS - RequireJS

- Keep JS modular
- Load the code you need, when you need it
- plugins like 'baja!' and module resources

```
require(['baja!', 'nmodule/js/rc/asyncUtils/asyncUtils'], function (baja,
asyncUtils) {
  baja.outln('bajascript is running.');
  return asyncUtils.waitInterval(3000)
    .then(() => {
        window.console.log("sleep some more");
     })
});
```

Run

Imports in JS - RequireJS

- Keep JS modular
- Load the code you need, when you need it
- plugins like 'baja!' and module resources

```
require(['baja!', 'nmodule/js/rc/asyncUtils/asyncUtils'], function (baja,
asyncUtils) {
  baja.outln('bajascript is running.');
  return asyncUtils.waitInterval(3000)
    .then(() =>
      windo
    })
});
```

```
bajascript is running.
```

Imports in JS - RequireJS

- Keep JS modular
- Load the code you need, when you need it
- plugins like 'baja!' and module resources

```
require(['baja!', 'nmodule/js/rc/asyncUtils/asyncUtils'], function (baja,
asyncUtils) {
  baja.outln('bajascript is running.');
  return asyncUtils.waitInterval(3000)
    .then(() =>
      windo
    })
});
```

```
bajascript is running.
sleep some more
```

RequireJS - loading types

```
require(['baja!', 'baja!alarm:AlarmClass'], function (baja, types) {

  window.console.log('type is ready: ' + baja.$('alarm:AlarmClass'));
});
```

Run

RequireJS - loading types

```
require(['baja!', 'baja!alarm:AlarmClass'], function (baja, types) {

  window.console.log('type is ready: ' + baja.$('alarm:AlarmClass'));
});
```

Run

```
type is ready: alarm:AlarmClass
```

Niagara Rpc

NF 23

TRIDIUM

Annotate to your existing Java methods.

```
@NiagaraRpc(
  permissions = "RWI",
  transports = {
    @Transport(type = fox),
    @Transport(type = box) //just add `box` Transport for Browser
  }
)
public String myFirstRpc(Context cx)
{
  return "Niagara JS Rules";
}
```

NiagaraRpc (fox)

```
  BFoxProxySession session = BFoxProxySession.make(someOtherHost, 4911,
/*useFoxs*/true, "aUserName", password);
  session.engageNoRetry("testSession");

  Optional response = session.rpc(
BOrd.make("station:|slot:/forum23/EchoComponent"), "myFirstRpc");

  if (response.isPresent()) {
    System.out.println(response.get());
  }
  session.disengage("testSession");
```

NiagaraRpc (box)

Simpler in BajaScript…

```
require(["baja!"], function(baja) {
  baja.Ord.make("station:|slot:/forum23/EchoComponent").get()
  .then(function (comp) {
    return comp.rpc("myFirstRpc");
  })
  .then(function (result) {
    console.log("The RPC response: " + result);
  });
});
```

Run

NiagaraRpc (box)

Simpler in BajaScript…

```
require(["baja!"], function(baja) {
  baja.Ord.make("station:|slot:/forum23/EchoComponent").get()
  .then(function (comp) {
    return comp.rpc("myFirstRpc");
  })
  .then(function (result) {
    console.log("The RPC response: " + result);
  });
});
```

```
The RPC response: Niagara JS Rules
```

`NiagaraRpc` Tips

- Methods can be static or instance based
- Always pass along the context for thing like BOrd resolution and BComponent.set calls
- Starting in Niagara 4.13, any BComplex results have switched from **`toString()`** to **`baja.bson.decodeAsync()`**.

Widget Model

NF 23

TRIDIUM

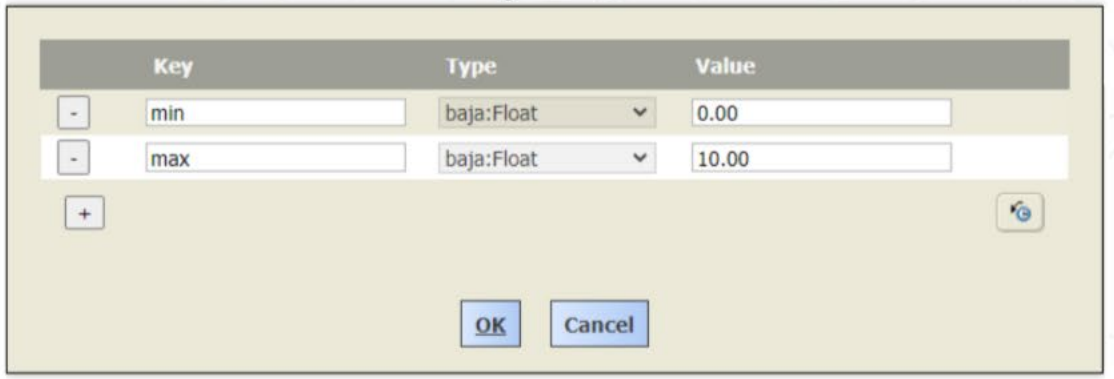
BWidget.java

- Builds on Component model
- BWbFieldEditor.java - small and for editing
- BPane.java - containers like BGridPane
- BWbView.java - Full screen Views

Widget.js

- BIFormFactors
  - BIFormFactorMini

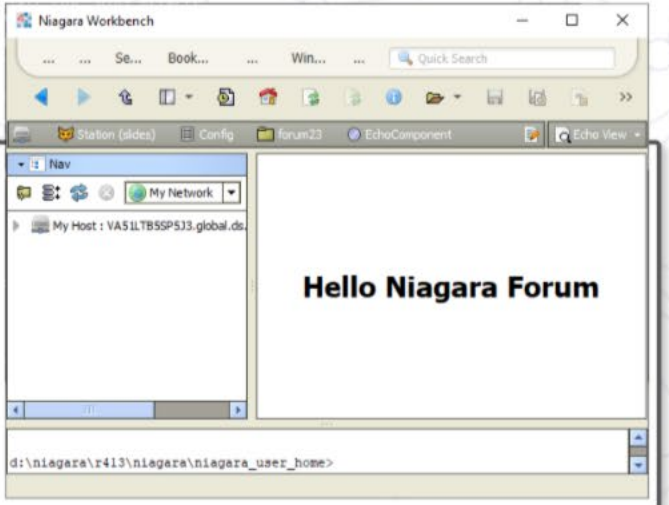
BIFormFactorCompact



- BIFormFactorMax
  - Full Screen Views
  - Show up in View Selector next to BWbViews

TRIDIUM

BWbView loading



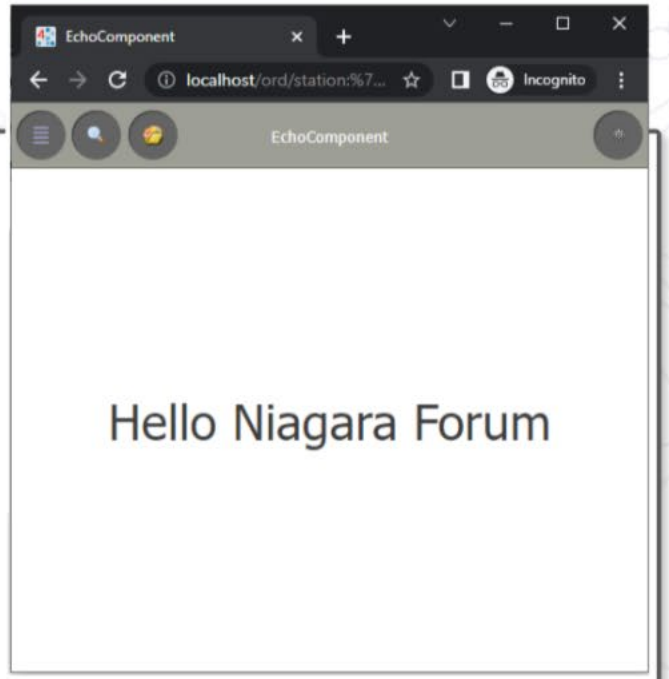

```
@NiagaraType(
  agent = @AgentOn(
    types = "slides:EchoComponent"
  )
)
public class BEchoView
  extends BWbView
{
  public void doLoadValue(BObject object, Context cx)
  {
    BEchoComponent component = ((BEchoComponent) object);
    setContent(new BLabel(component.toString()));
  }
}
```

BIFormFactor definition

```
@NiagaraType(
  agent = @AgentOn(
    types = "slides:EchoComponent"
  )
)
public class BEchoUxView extends BSingleton
  implements BIJavaScript, BIFormFactorMax
{
  @Override
  public JsInfo getJsInfo(Context cx) { return JS_INFO; }

  private static final JsInfo JS_INFO = JsInfo.make(
    BOrd.make("module://slides/rc/forum23/EchoView.js"));
}
```

Widget.js loading

```
class EchoView extends Widget {

  doInitialize(dom) {
    dom.html('<span></span>');
  }

  doLoad(echoComponent) {
    this.jq().children('span').text(echoComponent);
  }
}
```



Finding field editors Java vs JS

```
BString value = BString.make("hello");
BWbFieldEditor fieldEditor = BWbFieldEditor.makeFor(value);
fieldEditor.load(value);
```

```
require(['baja!', 'nmodule/webEditors/rc/fe/fe'], (baja, fe) => {

  const dom = $('#placeholder');
  const value = 'hello ux';
  return fe.buildFor({dom, value});
});
```

Run

- Placeholder -

Finding field editors Java vs JS

```
BString value = BString.make("hello");
BWbFieldEditor fieldEditor = BWbFieldEditor.makeFor(value);
fieldEditor.load(value);
```

```
require(['baja!', 'nmodule/webEditors/rc/fe/fe'], (baja, fe) => {

  const dom = $('#placeholder');
  const value = 'hello ux';
  return fe.buildFor({dom, value});
});
```

Run

hello ux

Dialogs and field editors Java vs JS

```
  BString value = BString.make("hello Java dialog");
  String result = BWbFieldEditor.dialog(parent, "Title", value);
```

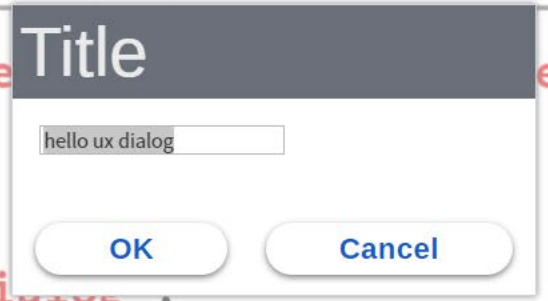
```
  require(['baja!', 'nmodule/webEditors/rc/fe/feDialogs'],
    (baja, feDialogs) => {

  const value = 'hello ux dialog';
  return feDialogs.showFor({value, title: 'Title'});
});
```

Run

Dialogs and field editors Java vs JS

```
BString value = BString.make("hello Java dialog");
String result = BWbFieldEditor.dialog(parent, "Title", value);
```

```
require(['baja!', 'nmodule                    eDialogs'],
  (baja, feDialogs) => {

  const value = 'hello ux dialog';
  return feDialogs.showFor({value, title: 'Title'});
});
```



Run

TRIDIUM

BPane.java - Combining Widgets made easy

- BGridPane.java Example

```
BGridPane gridPane = new BGridPane(1);
gridPane.add(null, new BLabel("First Row"));
gridPane.add(null, new BLabel("Second Row"));
```

bajaux/Spandrel

- Build widgets from other widgets
- Automatic re-renders
- With JSX, can build out widgets together

Combining Widgets made easy

```
1  /** @jsx spandrel.jsx */
2
3  const StringEditor = spandrel((string) => {
4    return <input type="text" value={string}/>;
5  });
6
7  return spandrel([
8    <StringEditor value={'Hello world!'} />
9  ]);
10
```

Hello world!

Combining Widgets made easy

```jsx
/** @jsx spandrel.jsx */

const StringEditor = spandrel((string) => {
  return <input type="text" value={string}/>;
});

return spandrel([
  <StringEditor value={'Hello world!'} />,
  <StringEditor value={'Hello world!'} />,
  <StringEditor value={'Hello world!'} />,
  <StringEditor value={'Hello world!'} />
]);
```

Commands

NF 23

TRIDIUM

Java - javax.baja.ui.Command

```
class EchoCommand extends Command{
  EchoCommand(BWidget owner){
    super(owner, Lexicon.make('slides'), "slides.cmds.echo");
  }

  public CommandArtifact doInvoke() {
    System.out.println("Java Command Complete");
  }
}
//Lexicon entry for Java Command
slides.cmd.echo.label=Echo Command
slides.cmd.echo.description=Echo Command Description
slides.cmd.echo.icon=modules://icons/x16/cloud.png
```

JS - bajaux/commands/Command

```
class EchoCommand extends Command {
  constructor() {
    super({
      module: 'slides',
      lex: 'slides.cmds.echo',
      func: => window.console.log('JS Command Complete')
    });
  }}
//Lexicon entry for JS Command
slides.cmd.echo.displayName=Echo Command
slides.cmd.echo.description=Echo Command Description
slides.cmd.echo.icon=modules://icons/x16/cloud.png
```

JS - Command Tips

- Starting in Niagara 4.14, you can even reuse existing lexicon entries in bajaScript:

```
slides.cmd.echo.label=Echo Command
slides.cmd.echo.displayName={lexicon:slides.cmd.echo.label}
```

Lexicons

NF 23

TRIDIUM

Java - Using Lexicons

```
  //For this lexicon entry in the alarm module
  //alarm.multipleAlarmNotes=<Multiple Alarms>
  Context cx = new BasicContext(user, "en")
  Lexicon lex = Lexicon.make("alarm");
  lex.getText("alarm.multipleAlarmNotes", cx); // \<Multiple Alarms\>;
  lex.getHtmlSafeText("alarm.multipleAlarmNotes", cx); // &amp;lt;Multiple
Alarms&amp;gt;
```

JS - lex plugin

```
require(["lex!alarm"], function (lexs) {
  const [ lex ] = lexs;
  console.log("Not safe: ", lex.get('alarm.multipleAlarmNotes'));
  console.log("Safe: ",     lex.getSafe('alarm.multipleAlarmNotes'));
});
```

Run

Lexicon Tips

- Watch out for double escaping,

```
require(['lex!alarm', 'dialogs'], function (lexs, dialogs) {
  const [ lex ] = lexs;
  dialogs.showOk({
      title: lex.get('alarm.multipleAlarmNotes'), //title right
      text: lex.getSafe('alarm.multipleAlarmNotes') //double escaped
  });
});
```

Run

Lexicon Tips

- Watch out for double escaping,

```
require(['lex!alarm', 'di                                    xs, dialogs) {
  const [ lex ] = lexs;
  dialogs.showOk({
      title: lex.get('ala                                    '), //title right
      text: lex.getSafe('alarm.multipleAlarmNotes') //double escaped
  });
});
```

<Multiple Alarms>

<Multiple Alarms>

OK

Run

Summary

- Introduced to a lot of similarities
- Niagara JS is always improving

Summary

- Thank you!
- Resources
  - **`grunt-init-niagara`** sets up everything
  - Tutorials in Doc Developer ➞ **`bajaux`** -Wb -> Help -> Doc Developer -> Building Javascript Applications