

Making Ultra Your Own

UEF for Beginners

13 July 2022

Chris Filkins – Collegis Education



What's on deck...



Introduction

- Who I am and what I do
- What Collegis does

Our History with Bb Development

- Building Blocks
- LTI Tools
- UEF Integrations

Ultra Extension Framework

- What is it?
- How to get started
- Example Code

A quick bit about me...

Previous lives...

- High school physics teacher
- Master teacher for physics education program
- Faculty support for data analytics in higher ed
- Faculty training through multiple LMS migrations
- Instructional Design for online courses
- LMS Administrator for multiple platforms
- Developed basic enhancements for LMS platforms (Nuggets for ANGEL)

Current life...

- Manager of Learning Applications
- Administer LMS platforms for multiple institutions
- Transition large institution from ANGEL to Bb Learn
- Developed 20+ building blocks to replace functionality from ANGEL in Bb Learn
- Transitioned multiple partners from self-hosted to SaaS
- Transitioned multiple partners to Ultra Base Navigation and Ultra Course View.
- Developed LTI tools to replace building blocks
- Developed UEF tools to provide requested custom features for clients.

EMPOWERING HIGHER EDUCATION TO
GROW ENROLLMENT AND REALIZE THE
PROMISE OF TECHNOLOGY



Enrollment Growth

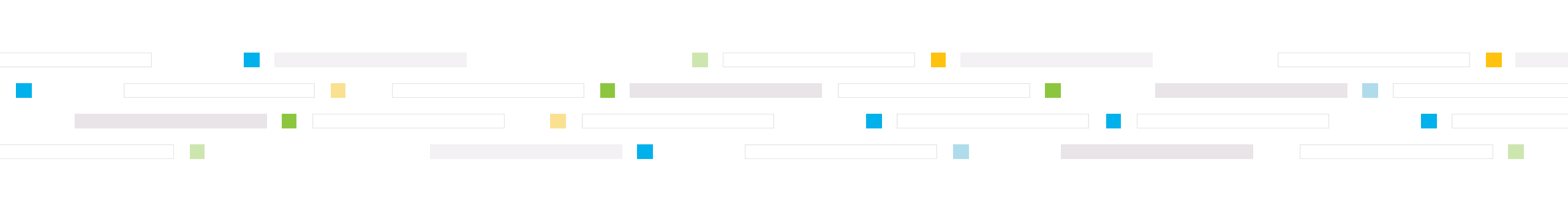


Data Enablement



Technology Management

- Empowering colleges and universities to grow enrollment and realize the promise of technology.
- 20+ years of working to support our partners in higher education.
- Assist our partners in areas including enrollment growth, anytime anywhere learning, and sustainable futures.



Bb Learn and Me

A brief history

Building Blocks

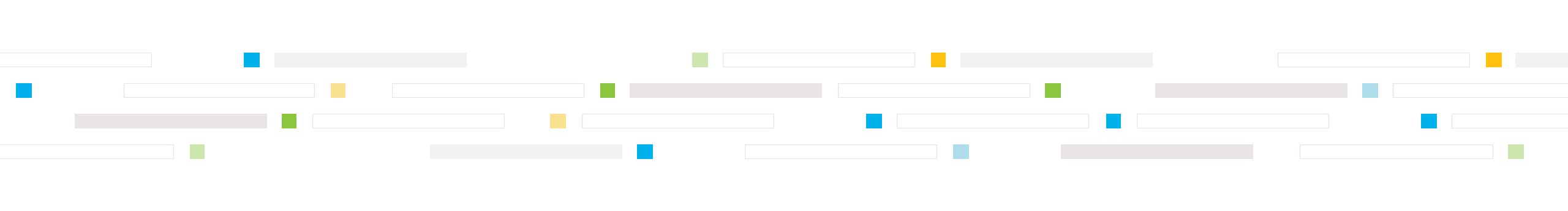
- Took advantage of every Technical Knowledge Training (TKT) program offered by Bb when we transitioned from ANGEL to Bb Learn
- Replicate functionality in ANGEL that was absent in Bb Learn
- Development was done using Spring Tool Suite and a local instance of Artifactory for our own Maven repositories.
- B2s included basic custom HTML modules, footers and banners, timers, custom quick enroll, custom login as, reporting tools, etc...20+ B2s deployed in our largest production instance

LTI Tools

- Partner decided to make switch to 100% Ultra courses
- Prioritized which B2s needed to be migrated to LTI tools
- Decided to use Tsugi (<https://www.tsugi.org/>) as our primary platform for hosting all of our LTI tools
 - Allows us to focus on the tools themselves and not worry about LTI configuration
- Began deploying LTI course and system tools in production as part of Ultra launch
 - Custom syllabus editor
 - Simple redirect tool
- Worked well for basic things, but our users wanted more integrations, and more of a native feel

Ultra Extension Framework tools

- Spiritual successor to jsHacks
- Set of Premium APIs that unlock a number of features and capabilities
 - Detailed telemetry data regarding user navigation – clicks, hovers, and routing
 - Ability to manipulate the user interface with the addition of pop-up notifications, modal windows, peek panels, banners, etc.
 - Leverage the Help tool area in Ultra to insert additional resources and/or tools
- Commercial examples include EesySoft, Watermark Course Evaluations & Surveys, Qwickly, eXplorance



Ultra Extension Framework

A shallow dip into a deep pool

Ultra Extension Framework basic mechanics

- UEF is an LTI tool with a specialized placement as an “Ultra tool”
- LTI launch loads a hidden iframe into Ultra base navigation
- Typically, iFrame handles authentication/authorization for REST API using 3LO
 - Be aware of security constraints around running code in an iframe, especially from an external service. Most modern browsers will block any cookies your tool relies on unless they are set to SameSite=none and Secure=true
- Auth token then gets passed to javascript, which establishes a messageChannel through which to communicate with Ultra

Saying hello

- Everything is done through a messageChannel
- Let's add in an event listener to pick up on what Bb's trying to tell us

```
window.addEventListener("message", onPostMessageReceived, false);
```

- We say howdy

```
window.parent.postMessage({  
  "type": "integration:hello"  
}, learn_url + '/*');
```

- When Bb responds, we create a logged messageChannel and then authorize our requests

```
messageChannel.postMessage({  
  type: 'authorization:authorize',  
  token: token  
});
```

Reading the room

- Once Bb responds that we're authorized, we can subscribe to telemetry data

```
messageChannel.postMessage({  
  type: 'event:subscribe',  
  subscriptions: ['click', 'hover', 'route', 'portal:new', 'portal:remove']  
});
```

- Depending on your tool, subscribe to as many (or as few) as needed
- The logged messageChannel and listener are very handy for watching your javascript console and identifying what information is available for each event, or the ids for any specific elements on a page.
 - It's worth noting that some telemetry data has more data than others, and depending on your tool, you may need to store the results.

Don't just stand there, do something!

- Add a help provider

```
messageChannel.postMessage({  
  displayName: "Enroll As...",  
  iconUrl: "https://iconhub.stuff/favicon.ico",  
  id: "collegis-enrollAs",  
  providerType: "auxiliary",  
  type: "help:register"  
});
```

Wait, what if I want to add some conditions?

- Telemetry data will typically be of the type event:event
- Keep in mind that a click into a course results in a number of events being fired
 - Click, Route, Portal:new
 - Route will contain the courseId (pk1), while portal:new will have values needed to inject custom content into a page

- Am I in a course now?

```
if (message.data.routeName.includes('peek.course.outline')) {...
```

- What course is this again?

```
var courseId = message.data.routeData.courseId;
```

Go ahead and inject some content

- Let's add a banner... (the portalId for the course page was stored when the page loaded)

```
messageChannel.postMessage({
  type: 'portal:render',
  portalId: sessionStorage.getItem('portalId'),
  contents: {
    tag: 'iframe',
    props: {
      style: {
        height: '60px',
        width: '100%'
      },
      src: panel_url + '&courseId=' + data.courseId + '&id=' + data.id
    }
  }
});
```

What about a peek panel?

- Panels are a little more work...start by defining the panel

```
setTimeout(() => {  
  messageChannel.postMessage({  
    type: 'portal:panel',  
    correlationId: 'enrollAsPanel-1',  
    panelType: 'small', // full is the other option  
    panelTitle: 'Enroll As...',  
    attributes: {  
      onClose: {  
        callbackId: 'enrollAsPanel-1-dose'  
      }  
    },  
  });  
}, 2000);
```


Render the panel

Ultra will return an ID for the new panel...

```
if (message.data.type === 'portal:panel:response')
{
  renderPanelContents(message);
}

{...}
```

...and we'll submit a request to render our iframe in that panel

```
panelId = message.data.portalId;
messageChannel.postMessage({
  type: 'portal:render',
  portalId: message.data.portalId,
  contents: {
    tag: 'span',
    children: [{
      tag: 'iframe',
      props: {
        style: {
          flex: '1 1 auto'
        },
        src: panel_url + '&course_id=' + course_id + '&action=' + action
      }
    ]
  }
});
```

Additional injection points

- Notifications

- Notifications are similar to panels in general workflow – request a notification, Bb respond with an id, render the content into that id
- Notifications are anchored to other items on the page or to a region

```
messageChannel.postMessage({
  type: 'portal:notification', // note that there's no manual id assignment here
  selector: { // notifications can appear anchored to an item on the page, or in a region
    type: 'notification:analytics:selector',
    value: 'panel.header.open.settings.panel.link'
  },
  // selector: {
  //   type: 'notification:region:selector',
  //   value: 'bottom:right'
  // },
  contents: 'Blah'
});
```

Course Details

- Allows the addition of items under the Details & Actions menu within an Ultra Course
 - Tool needs to be registered after script is authorized

```
messageChannel.postMessage({  
  "type": "course:detail:register",  
  "registrationName": tool_name  
});
```

- Ultra will respond with a registrationId for the tool
 - Unlike other tools, you can register multiple times, so it's important to listen for this specific registrationId
- ```
if (message.data.selector === "course.outline.details" && message.data.selectorData && message.data.selectorData.registrationId === localStorage.getItem(tool_name + '_registrationId')) {
 insertTool(message.data.portalId, courseId);
}
```
- From here, you render the content and links for the tool.

# Tips we've picked up along the way

- Browser security can make some things challenging
  - Passing data between Ultra and UEF tool can be done via localStorage as a medium
    - sessionStorage has different security settings and doesn't work for cross-site purposes
  - Add a listener for storage events, and retrieve/clean up data as it's being saved from different sources
- Watch the console closely to observe all of the telemetry data for a given workflow
  - Lots of things can trigger a portal:new within a course
- Keep track of your requests!
  - UEF tools can consume a tremendous number of requests
  - Cache whatever data you safely can

# More tips

- Be security conscious!
  - Don't save anything in localStorage or sessionStorage you don't want users to get access to
  - Handle the OAuth authentication server-side and only store the token for 3LO
    - If you develop a tool that requires elevated permissions beyond what the user has, handle all of those transactions server-side and do not expose that token!
- Have fun and be creative!

# Resources

- Collegis Education Github repo
  - <https://github.com/CollegisEduLTS>
- Blackboard Dev Docs – Getting started with UEF
  - <https://docs.blackboard.com/rest-apis/learn/uef/getting-started>
  - <https://github.com/blackboard/BBDN-UEF-Python>
- Tsugi Project
  - <https://github.com/tsugiproject/tsugi>

# Thanks for coming!

Chris Filkins

[chris.Filkins@collegiseducation.com](mailto:chris.Filkins@collegiseducation.com)

(630)210-3481