

Introduction

ASSISTments provides the Problem type, *Externally Run (in an iframe)*, which allows you, a researcher, to integrate externally created learning resources into an ASSISTments Problem Set. Examples of including content from external projects can be seen in these Problem Set Test Drives:

- AutoTutor ([Test Drive](#))
- BBN's ElectronixTutor ([Test Drive](#))
- Dragoon ([Test Drive](#))
- Jiune PSABWZYK

To have your learning resource embedded via an iframe in an ASSISTments problem, your learning resource must be made *ASSISTments-aware*. An *ASSISTments-aware* learning resource posts a set of ASSISTments-specified messages from your iframe URL back to assistments.org.

ASSISTments expects to receive these messages from your iframe's learning resource:

1. loaded - An initial message tells assistments.org your server has successfully loaded its content in the iframe. assistments.org expects to receive this within <tbs> seconds.
2. heartbeat - A repeating messaging sent at a recommended interval of every 20 seconds indicates your content is alive and well. If assistments.org does not receive this message within 30 seconds of the last heartbeat message, assistments.org assumes your iframe has ceased working and presents an error message to the student.
3. completed - At the conclusion of student activity, your iframe sends this message, along with a student score and answer:
 - a. score - Required: a value between 0.0 and 1.0
 - b. answer - Optional: plain text or an html fragment representing the student's response.

Both of these values appear in the assignment's report that ASSISTments provides to its users, students and teachers.

Two approaches are available for making your learning resource ASSISTments-aware:

- a. Integrate your own, minimal Javascript into your iframe content that sends the necessary messages to assistments.org. A simple example of this approach is shown next under "Step by Step"
- b. Take advantage of [SuperGLU](#), a rich set of Javascript utilities for communicating data between different learning technologies and intelligent systems. [This tutorial](#) shows how to use SuperGLU.

Step by Step

Integrating your external content with assistments.org involves these steps:

1. To enable the *Externally Run (in an iframe)* feature:
 - Log into ASSISTments and click "Preferences" in the blue menu bar
 - Open the "Builder Settings" accordion
 - Click "[+] Beta/Research Settings"
 - Turn "Show External Problem Types" to "On" (see highlighted image.)

- Click “Save Builder Settings”

[\[-\] Beta/Research Settings](#)

On Off

- ☒ ☐ Show attempts count option when creating you
- ☒ ☐ Show "Assemble Problem Set Using Existing Pr
- ☐ ☒ Show "Create Skill Builder Problem Set" link (B
- ☒ ☐ Show "Skills" link
- ☒ ☐ Show "Problems" link
- ☒ ☐ Can add sections to problem sets
- ☒ ☐ Show advanced skill builder settings
- ☒ ☐ Show "Share Links" link (Details page)
- ☒ ☐ Creates only textbook problems.
- ☒ ☐ Can create placements problem sets
- ☒ ☐ **Show External Problem Types**
- ☐ ☐ Show Variabilized Problem Types Options

Save Builder Settings

Restore Defaults

In the Builder, you will now have available the Problem type *Externally Run (in an iframe)*.

2. Register an email contact and the domain name for your server [here](#). ASSISTments loads only learning resources (in an iframe) that come from your registered URL. When the Tutor detects an error in your learning resource, the Tutor displays the registered email as the contact point for end users.

URL:

Contact Email:

☐ I would like to receive student names

☐ I have read and accepted the [Terms of Use for Data](#)

Please note: For your learning resources to seamlessly display in the Tutor, your server *must* be running over https.

3. Create a Problem Set containing an *Externally Run (in an iframe)*:
 - On the Builder page, click “Create Your Own Problem Set”
 - Edit the “Title” as you wish
 - For “Question 1”, select the Problem Type "*Externally Run*" as shown last on the list below

 [Save Problem Body](#)

Answer Type:
[\[-\] Numeric and Algebraic Answer Types](#)
☐ Number 
☐ Exact Fraction 
☐ Numeric Expression 
☐ Algebraic Expression 

[\[-\] Text Answer Types](#)
☐ Exact Match (case sensitive) 
☐ Exact Match (ignore case) 
☐ Ungraded Open Response 

[\[-\] Multiple Answer Types](#)
☐ Multiple Choice 
☐ Check All That Apply 
☐ Ordering 

[\[-\] Advanced Types](#)
☐ Write A Racket Problem 
☒ Externally Run 

- Click the Source Code button in the Question editor: 
- In the Source code pop up, enter the html for your Problem, including an iframe that loads your learning resource.

For example:

```
<p>
Problem body below is in an iframe loading content provided by an
external (non-assistments.org) server:
</p>
<iframe src="https://imladris.wpi.edu/astest/minimalChildWindow.html"
width="400" height="200"/>
```

Most likely you would want to include parameters on your URL, for instance to specify which particular learning resource to load.

The above Problem can be found in PSAWF29. Here you can [Test Drive](#) it on www.assistments.org. You can also [Test Drive](#) a version of the Problem Set on test1: PSAWF28.

4. The Tutor automatically appends the additional parameters to your iframe URL:
 - p1= User Reference (uuid)
 - p2= Class Reference (uuid)
 - p3= Assignment Reference (uuid)
 - p4= Problem ID Decoded (id) [formerly called "Assistment ID" in this doc]
 - p5= Question ID (id) [formerly called "Problem ID" in this doc]
 - p6= User Name (if your domain registration request for this item was approved)
 - p7= ASSISTments System Reference (uuid)

Your server can optionally associate the data your learning resource records with the above ASSISTments-provided information.

5. [Open the above iframe src URL](#) and view its page source. You will see it contains Javascript functions that makes the iframe page ASSISTments-aware by posting the required 3 JSON formatted messages to assistments.org:
 - `iframeLoaded()` posts:

```
{ "version" : "AS.1", "action" : "loaded" }
```
 - `iframeHeartbeat()` posts:

```
{ "version" : "AS.1", "action" : "heartbeat" }
```
 - `iframeCompleted()` posts (assuming you entered the value: 0.21):

```
{ "version" : "AS.1", "action" : "completed", "answer" : "student entered: 0.21", "score" : 0.21 }
```

Note: The “answer” field and value are optional. The value of the answer is completely up to your learning resource to define.
- “AS.1” tells the Tutor what data and format to expect in the JSON message. At this time “AS.1” is the sole supported message version.
6. At the conclusion of running the Problem Set, a Student’s report shows as “My Answer” whatever value your learning resource returned in the “completed” message:

3 - Minimal Case: Externally Run Problem (in an iframe)

Problem	My Answer	My Classmates' Average
PRA6Q45	✖ Student worked hard on this problem.	0%

My average: 0%

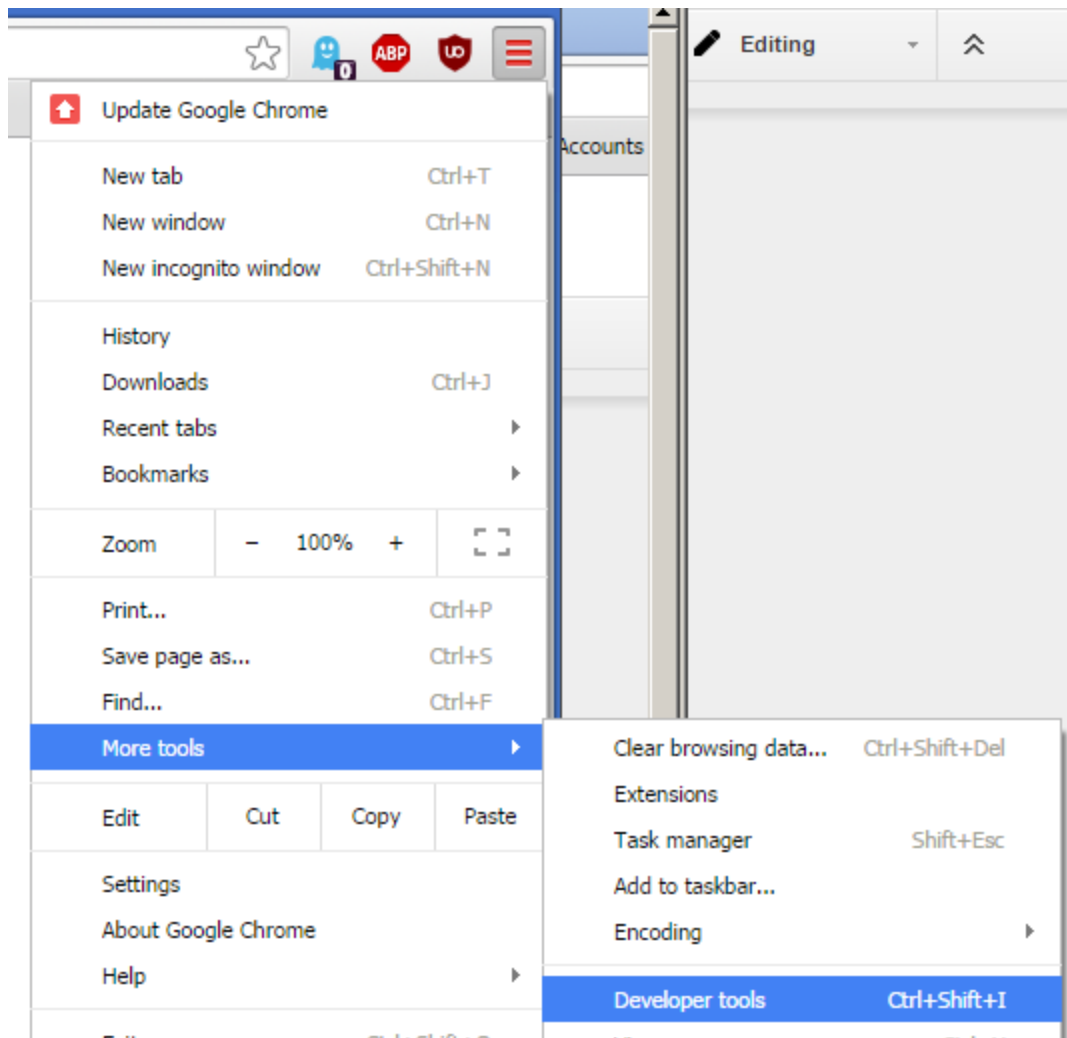
Similarly, the Teacher’s report shows the same information:

✖ Student worked hard on this problem.

Debugging Tips

Using Chrome here are some tips for debugging your iframe Problems:

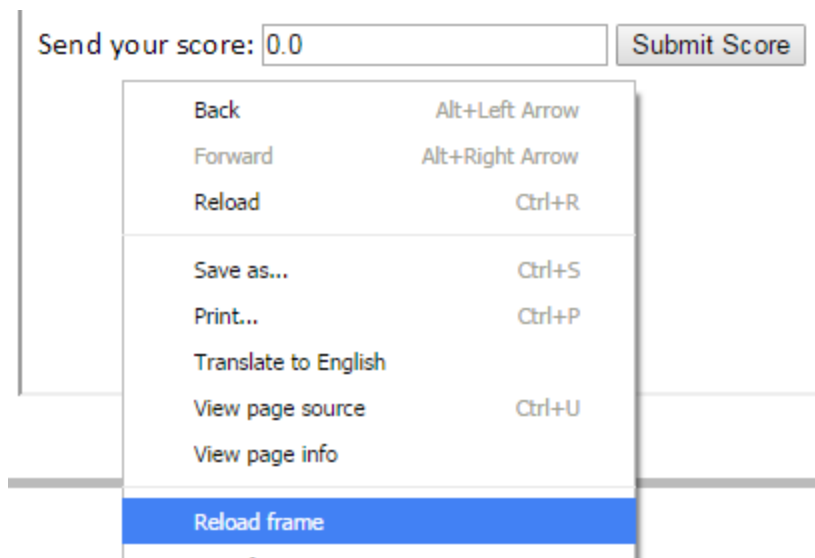
1. Go to your Problem in the Builder
2. Open the “Test Drive” link in a new tab
3. Switch to that tab
4. Open Chrome’s Developer tools with the shortcut `Ctrl+Shift+I`, or use the menu:



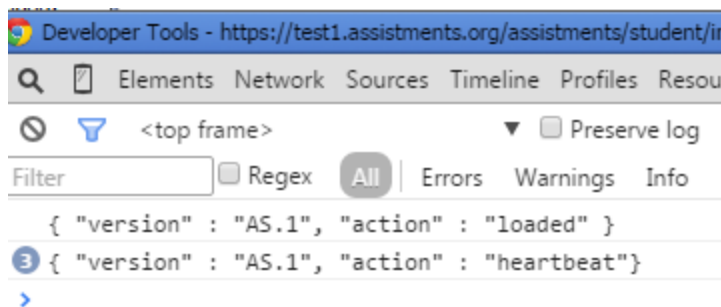
5. In the Developer tools frame, click the “Console” tab and enter this command:
`window.addEventListener("message", function(e){console.log(e.data)})`

That will cause the messages posted by your iframe page to be echoed in the console window.

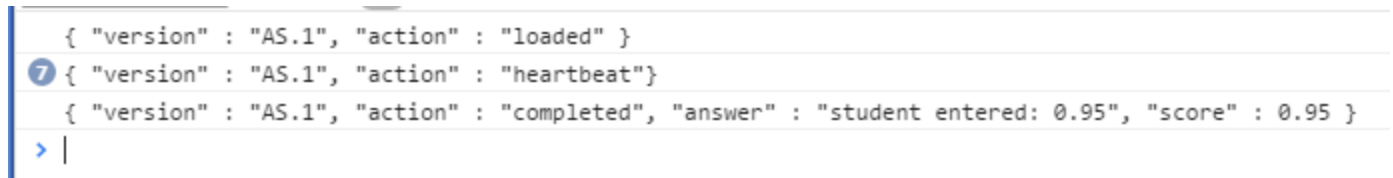
6. Back on the Test Drive tab, reload your iframe by right-mouse clicking on the iframe and selecting “Reload frame”:



7. Now you will see your loaded message and repeating heartbeats until you submit a score:

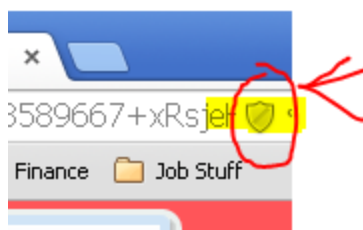


8. Finally the score and optional answer:

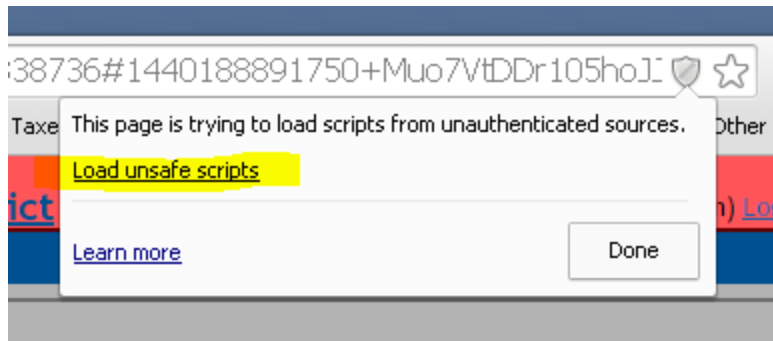


9. If your page contains Javascript errors, those errors will be shown in the console window and clicking the line number brings the error up in the Developer tools frame.
10. If your server does not currently run with https, when the Tutor loads your iframe problem, you learning resource does not initially display. You must tell the browser that it is OK to load insecure content (from your server) in the context of the secure assistments.org page.

Using Chrome as an example, click the shield in the upper right corner of the URL bar:



The circled, greyed-out shield is warning about the http content. By clicking the icon, you'll see:



Click "Load unsafe scripts" and the page will reload, showing you the iframe content.

Contact Us:

If you have suggestions for improving this page or need additional help, please contact us at assistments-help@wpi.edu

Last updated: 10/2/2015