AdaWebPack

Getting started

FOSDEM 2022

Max Reznik

As an Ada evangelist he supports xUSSR Ada community since 2002 by developing a web site, forum/mail group, Telegram. Lot of Ada repositories

on GitHub.

https://www.ada-ru.org

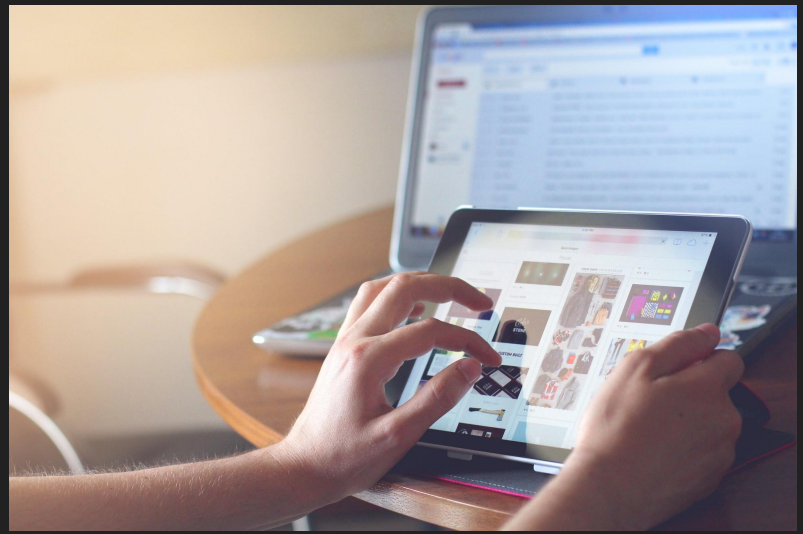
https://github.com/reznikmm

What is AdaWebPack?

The toolkit enables Ada in Web Application development.

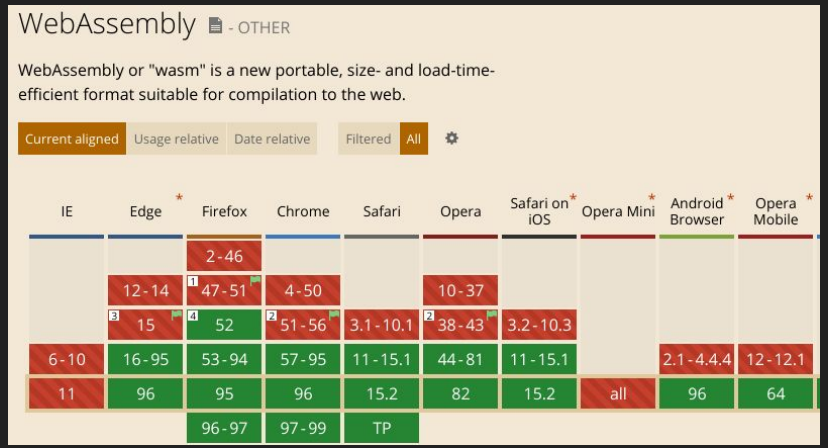
It includes:

- WebAssembly toolchain
- Customized Ada Runtime Library
- Web API binding



WebAssembly

- Portable binary format
- VM embedded into browser
- Language independent



LLVM

- IR
- Many frontends
  - incl. Ada
- Many backends
  - incl. WebAssembly

GNAT LLVM

- Joins GNAT frontend with LLVM backend

  ➢ `./llvm-gcc -c hello.adb`

- Included into **gprbuild** knowledge DB

  ➢ `gprbuild --target=llvm hello.adb`

How to compile Ada to WebAssembly

- Pass `--target=wasm32` LLVM compiler switch
- Point GNAT to target dependent information
  - `-gnateT=wasm32.atp`
- Pass LLVM target to `gprbuild`
  - `gprbuild --target=llvm`

```
llvm-gcc -c --target=wasm32 -gnateT=wasm32.atp hello.adb
```

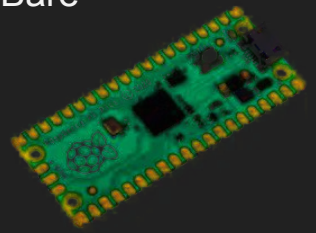
```
gprbuild --target=llvm -P hello.gpr -cargs \
    --target=wasm32 -gnateT=wasm32.atp
```

```
1   Bits_BE                         0
2   Bits_Per_Unit                   8
3   Bits_Per_Word                  32
4   Bytes_BE                        0
5   Char_Size                       8
6   Double_Float_Alignment          0
7   Double_Scalar_Alignment         0
8   Double_Size                    64
9   Float_Size                     32
10  Float_Words_BE                  0
11  Int_Size                       32
12  Long_Double_Size               64
13  Long_Long_Size                 64
14  Long_Size                      32
15  Maximum_Alignment              16
16  Max_Unaligned_Field            64
17  Pointer_Size                   32
18  Short_Enums                     0
19  Short_Size                     16
20  Strict_Alignment                0
21  System_Allocator_Alignment     16
22  Wchar_T_Size                   32
23  Words_BE                        0
24
25  float          6  I  32  32
26  double        15  I  64  64
27  long double   15  I  64  64
```

Ada Run-Time Library

~~GNAT Run-Time based on POSIX doesn't match WebAssembly~~

Custom Runtime based on Bare Board Runtime



Restrictions

- No exceptions (but local)
- No nested subprogram access
- No tasks and protected objects

```
40  pragma Restrictions (No_Exception_Propagation);
41  --  Only local exception handling is supported in this profile
42
43  --  pragma Restrictions (No_Exception_Registration);
44  --  Disable exception name registration. This capability is not used because
45  --  it is only required by exception stream attributes which are not supported
46  --  in this run time.
47
48  pragma Restrictions (No_Implicit_Dynamic_Code);
49  --  Pointers to nested subprograms are not allowed in this run time, in order
50  --  to prevent the compiler from building "trampolines".
51
52  pragma Restrictions (No_Tasking);
53  --  Tasking is not supported in this run time
54
55  pragma Restrictions (No_Task_Hierarchy);
56  --  Disable masters
57
58  pragma Restrictions (No_Abort_Statements);
59  pragma Restrictions (Max_Asynchronous_Select_Nesting => 0);
60  --  Disable abort deferral
61
62  pragma Restrictions (No_Initialize_Scalars);
63  --  To simplify the generated code
64
65  package System is
```

Custom Memory Allocator

- Based on TLSF allocator
- Request host to grow heap

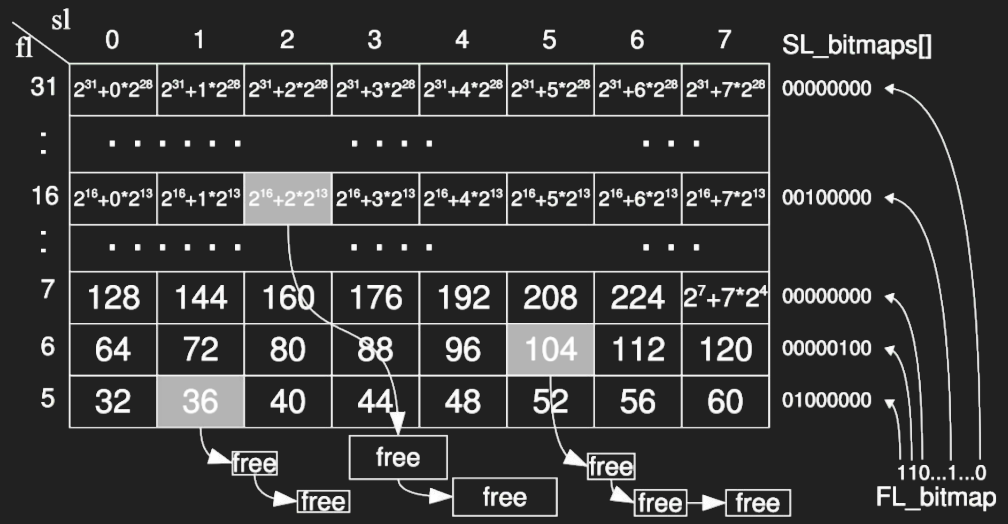
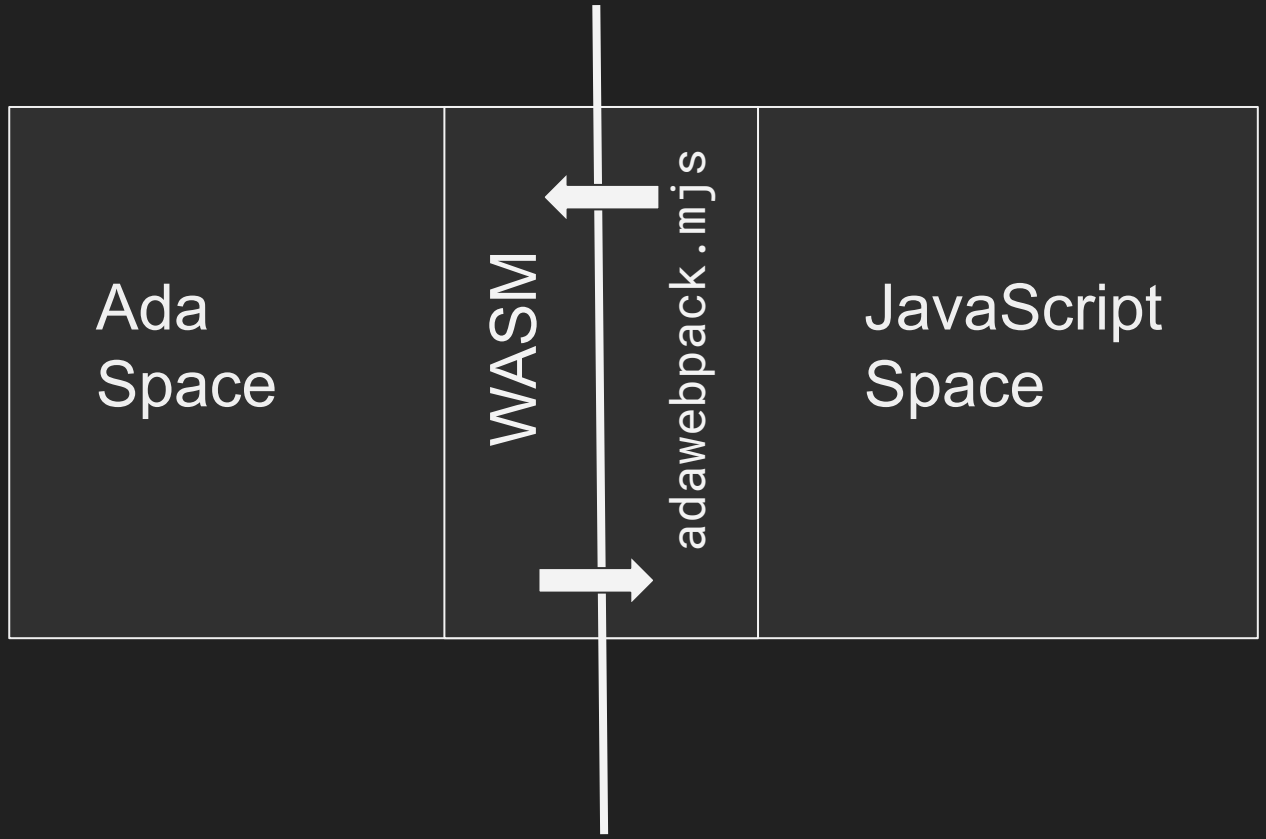


Fig. 1 TLSF data structures example

WASM Core packages

- `WASM.Objects`
  handles JS object references
- `Methods`
  calls JS object methods
- `Attributes`
  gets/set JS object attributes



Web API bindings – Web.Strings

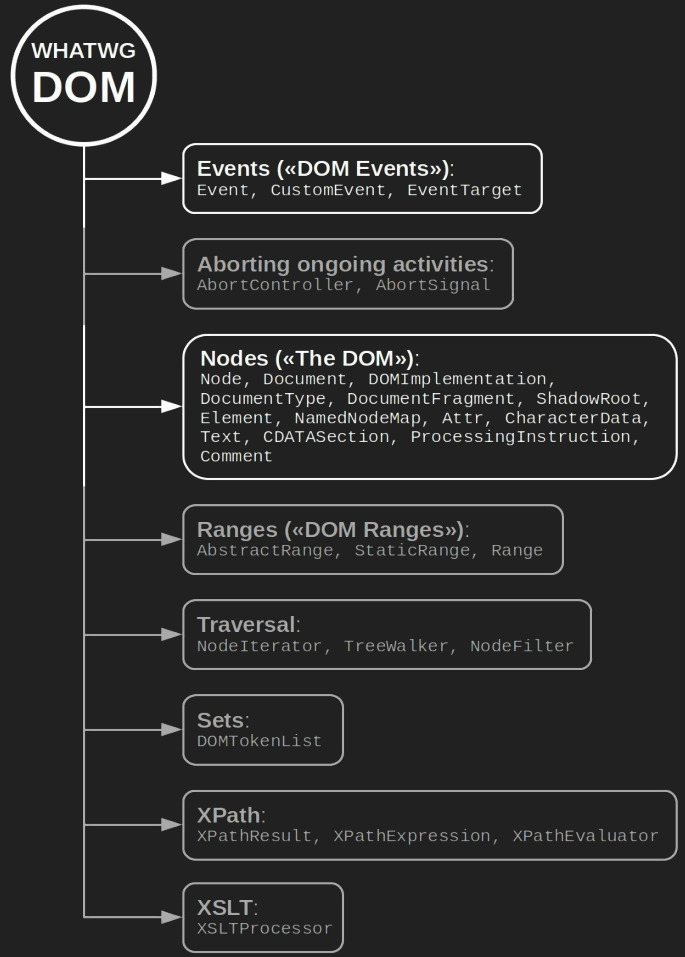
`Web.Strings.Web_String`

to exchange strings between Ada and JS

```
69
70      function "+" (Item : Wide_Wide_String) return Web.Strings.Web_String
71        renames Web.Strings.To_Web_String;
72
73      X : Web.HTML.Elements.HTML_Element
74        := Web.Window.Document.Get_Element_By_Id (+"toggle_label");
75
76
```

Web API bindings – Web.DOM

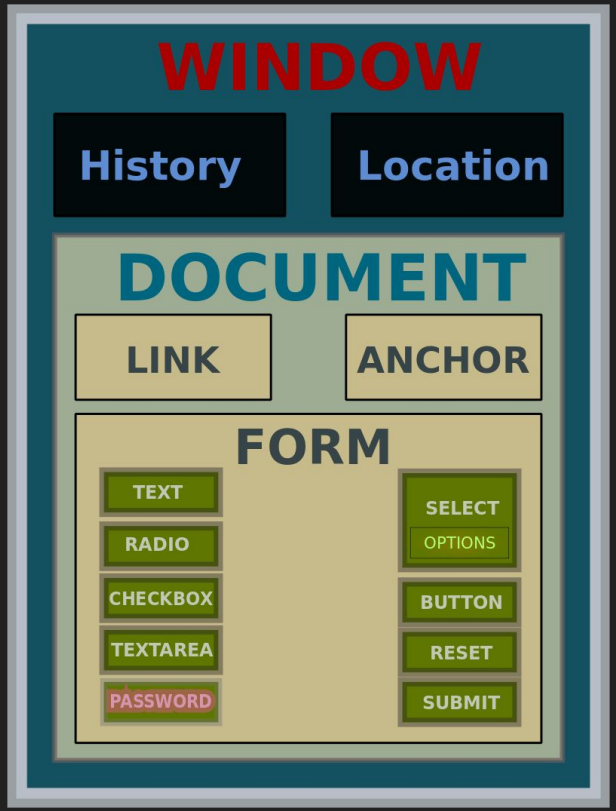
- Nodes
- Documents
- Elements
  - Get_Element_By_Id
- Events
- Event_Targets
- Event_Listeners
  - Add_Event_Listener



Web API bindings – Web.HTML

- Elements
- Buttons
- Inputs
- …
- Web.Window

```
42  procedure Initialize_Demo is
43     B : Web.HTML.Buttons.HTML_Button_Element
44       := Web.Window.Document.Get_Element_By_Id
45          (+"toggle_button").As_HTML_Button;
46
47  begin
48     B.Add_Event_Listener (+"click", L'Access);
49     B.Set_Disabled (False);
50  end Initialize_Demo;
```



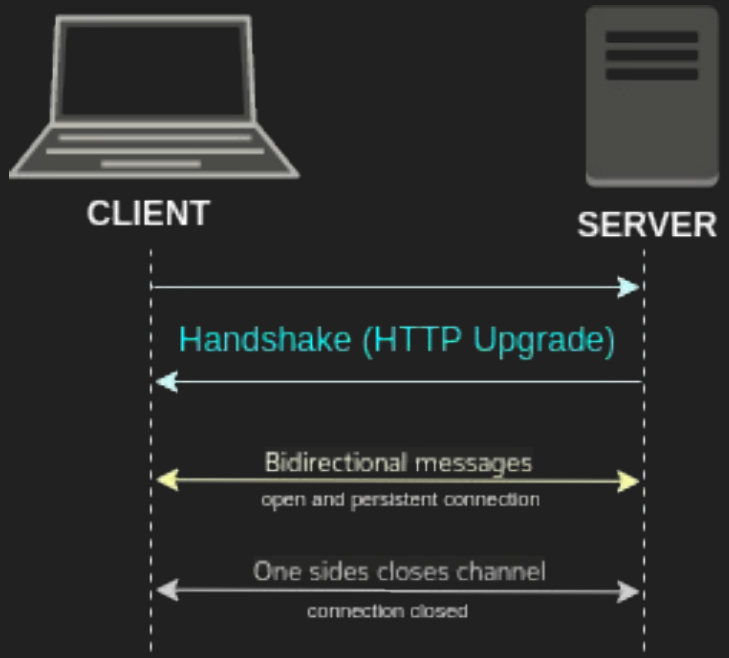
Web API bindings – Web.GL

WebGL - API to render 2D and 3D graphics

- Objects
- Buffers
- Shaders
- Textures
- …

Web API bindings – Web.Sockets, Web.XHR

- WebSocket - bidirectional protocol
- XHR - XMLHttpRequest



Toggle Hidden demo (1/5)

index.html ●

examples > toggle_hidden > index.html > { } "index.html"

```
1   <!DOCTYPE html>
2   <html>
3   >  <head> ···
17     </head>
18     <body>
19       <p id='toggle_label' hidden='hidden'>Hello, World!</p>
20       <button id='toggle_button' disabled='disabled'>Hello, Ada</button>
21     </body>
22   </html>
```

Toggle Hidden demo (2/5)

```
14    type Listener is
15      limited new Web.DOM.Event_Listeners.Event_Listener with null record;
16
17    overriding procedure Handle_Event
18     (Self  : in out Listener;
19      Event : in out Web.DOM.Events.Event'Class)
20    is
21       X : Web.HTML.Elements.HTML_Element
22         := Web.Window.Document.Get_Element_By_Id (+"toggle_label");
23
24    begin
25       X.Set_Hidden (not X.Get_Hidden);
26    end Handle_Event;
```

Toggle Hidden demo (3/5)

```
32    procedure Initialize_Demo is
33       B : Web.HTML.Buttons.HTML_Button_Element
34         := Web.Window.Document.Get_Element_By_Id
35            (+"toggle_button").As_HTML_Button;
36
37    begin
38       B.Add_Event_Listener (+"click", L'Access);
39       B.Set_Disabled (False);
40    end Initialize_Demo;
```

```
19       <p id='toggle_label' hidden='hidden'>Hello, World!</p>
20       <button id='toggle_button' disabled='disabled'>Hello, Ada</button>
```

Toggle Hidden demo (4/5)

```
1   package Demo is
2
3      procedure Initialize_Demo;
4
5   end Demo;
```

```
1   with Demo;
2
3   procedure Main is
4   begin
5      null;
6   end Main;
```

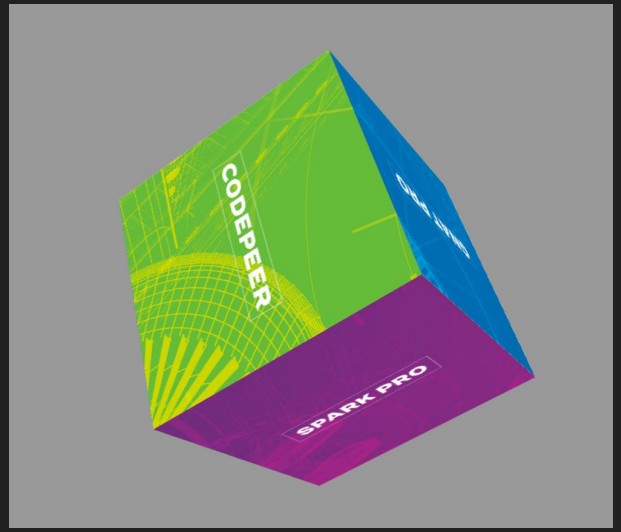
```
9   package body Demo is
10
11  >  function "+" (Item : Wide_Wide_String) return Web.Strings.Web_String ···
13
    >  type Listener is ···
       >  overriding procedure Handle_Event ···
       >  procedure Initialize_Demo is ···
    begin
       Initialize_Demo;
40  end Demo;
```

Toggle Hidden demo (5/5)

```
6       <script type='module'>
7         import * as AdaWebPack from './adawebpack.mjs';
8
9         (async () => {
10          const fetchPromise = fetch('main.wasm');
11          const { instance } =
12            await WebAssembly.instantiateStreaming(fetchPromise, {env: AdaWebPack.imports});
13          AdaWebPack.initialize(instance);
14          instance.exports['_ada_main']();
15        })();
16      </script>
17    </head>
18    <body>
```

Derived works – AdaGL, AdaWebUI, Garlic

- https://github.com/godunko/adagl
  - A common API with native and WebGL implementations
- https://github.com/godunko/adawebui
  - Widget toolkit
- https://github.com/reznikmm/garlic
  - Annex E/DSA for seamless client/server interaction



Questions or Ideas?