

Using Web Sequence Diagrams with your APIs

Mike Amundsen

API Academy

@mamund



The WebApp

The Node CLI app

web sequence diagram generator

Edit

Manage topics

8 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Table with 3 columns: File Name, Commit Message, and Time Ago. Rows include .gitignore, README.md, index.js, onboardingAPI.png, onboardingAPI.wsd, package-lock.json, and package.json.

```
#!/usr/bin/env node

/*
 * wsdgen utility
 * converts valid WSD file into PNG
 * calls http://www.websequencediagrams.com
 * 2018-06
 * mamund
 */

"use strict";

var wsd = require('websequencediagrams');
var program = require('commander');
var fs = require('fs');
var diag = {text:"",type:"",style:"",file:""};

// top-level routine
program
  .arguments('<file>')
  .action(function(file){wsdgen(file)})
  .parse(process.argv);

// do the work
function wsdgen(file) {
  var text;

  diag.file = file;
  diag.type = "png";
  diag.style = "vs2010";
  if(readWSD(diag)===true) {
    writeWSD(diag);
  }
}
```

```
// do the work
function wsdgen(file) {
  var text;

  diag.file = file;
  diag.type = "png";
  diag.style = "vs2010";
  if(readWSD(diag)===true) {
    writeWSD(diag);
  }
}

// read the wsd
function readWSD(diag) {
  var rtn = "";

  if(fs.existsSync(diag.file)) {
    diag.text = fs.readFileSync(diag.file,'utf8');
    rtn = true;
  }
  else {
    console.log("File not found: "+diag.file);
    rtn = false;
  }

  return rtn;
}

// write the png
function writeWSD(diag) {
  var text = ((diag.text&&diag.text!="")?diag.text:"missing WSD");
  var style = ((diag.style&&diag.style!="")?diag.style:"vs2010");
  var type = ((diag.type&&diag.type!="")?diag.type:"png");
  var outfile = ((diag.file&&diag.file!="")?diag.file.replace('.wsd','.png'):"wsdgen.png");

  wsd.diagram(text, style, type, function(err, buf){
    if(err) {
      console.error(err);
    }
    else {
      console.log('writing '+outfile);
      fs.writeFileSync(outfile, buf);
    }
  });
}
```

The Process

Map Resources

Map Resources

home->+WIP:

WIP->+customerData:

customerData-->-WIP:

WIP->+accountData:

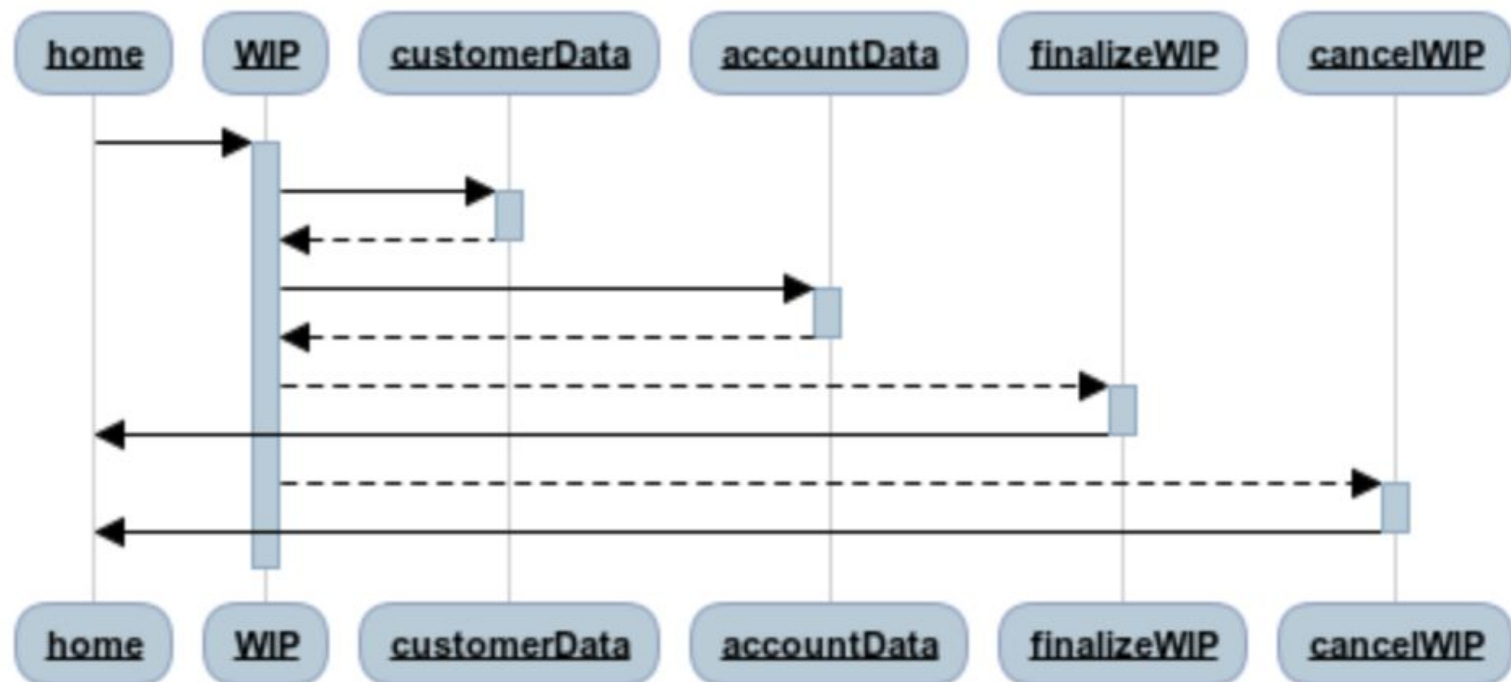
accountData-->-WIP:

WIP-->+finalizeWIP:

finalizeWIP->-home:

WIP-->+cancelWIP:

cancelWIP->-home:



Add Transitions

Add Transitions (hypermedia)

home->+WIP: startOnboarding

WIP->+customerData: collectCustomerData

customerData-->-WIP: saveToWIP

WIP->+accountData: collectAccountData

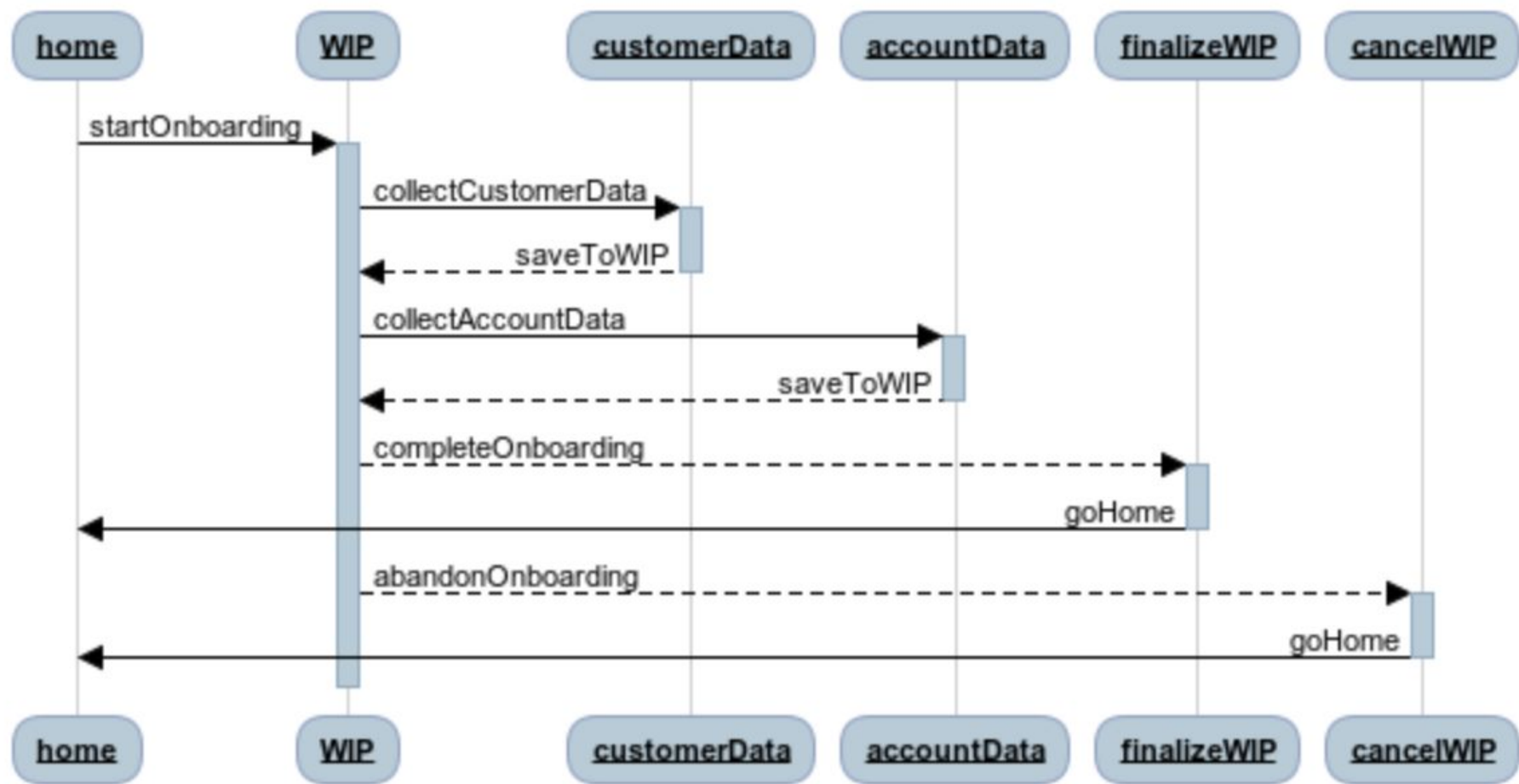
accountData-->-WIP: saveToWIP

WIP-->+finalizeWIP: completeOnboarding

finalizeWIP->-home: goHome

WIP-->+cancelWIP: abandonOnboarding

cancelWIP->-home: goHome



Add Parameters

Add Parameters (changing state)

```
home->+WIP: startOnboarding(identifier)
WIP->+customerData: collectCustomerData(identifier, name, email)
customerData-->-WIP: saveToWIP(identifier, name, email)
WIP->+accountData:
collectAccountData(identifier, region, discount)
accountData-->-WIP: saveToWIP(identifier, region, discount)
WIP-->+finalizeWIP: completeOnboarding(identifier)
finalizeWIP->-home: goHome
WIP-->+cancelWIP: abandonOnboarding(identifier)
cancelWIP->-home: goHome
```




Where can this lead us?

Generating Descriptions

```
{
  "alps": {
    "version": "1.0",
    "title": "onboardingAPI.wsd",
    "doc": {
      "type": "text",
      "value": "ALPS document for onboardingAPI.wsd"
    },
    "descriptors": [
      {
        "id": "startOnboarding",
        "type": "safe",
        "rtn": "",
        "descriptors": [
          {
            "href": "#identifier"
          }
        ]
      },
      {
        "id": "collectCustomerData",
        "type": "safe",
        "rtn": "",
        "descriptors": [
          {
            "href": "#identifier"
          },
          {
            "href": "#name"
          },
          {
            "href": "#email"
          }
        ]
      }
    ]
  }
}
```

Generating Documentation

On-Boarding API

ALPS Profile for BigCo's Customer On-Boarding API

Actions

The following actions, or state transitions, are defined for this API.

id	type	rt	args	notes
startOnboarding	safe	na	#identifier	
collectCustomerData	safe	na	#identifier , #name , #email	
saveToWIP	unsafe	na	#identifier , #name , #email	
collectAccountData	safe	na	#identifier , #region , #discount	
saveToWIP	unsafe	na	#identifier , #region , #discount	
completeOnboarding	unsafe	na	#identifier	
goHome	safe	na		
abandonOnboarding	unsafe	na	#identifier	

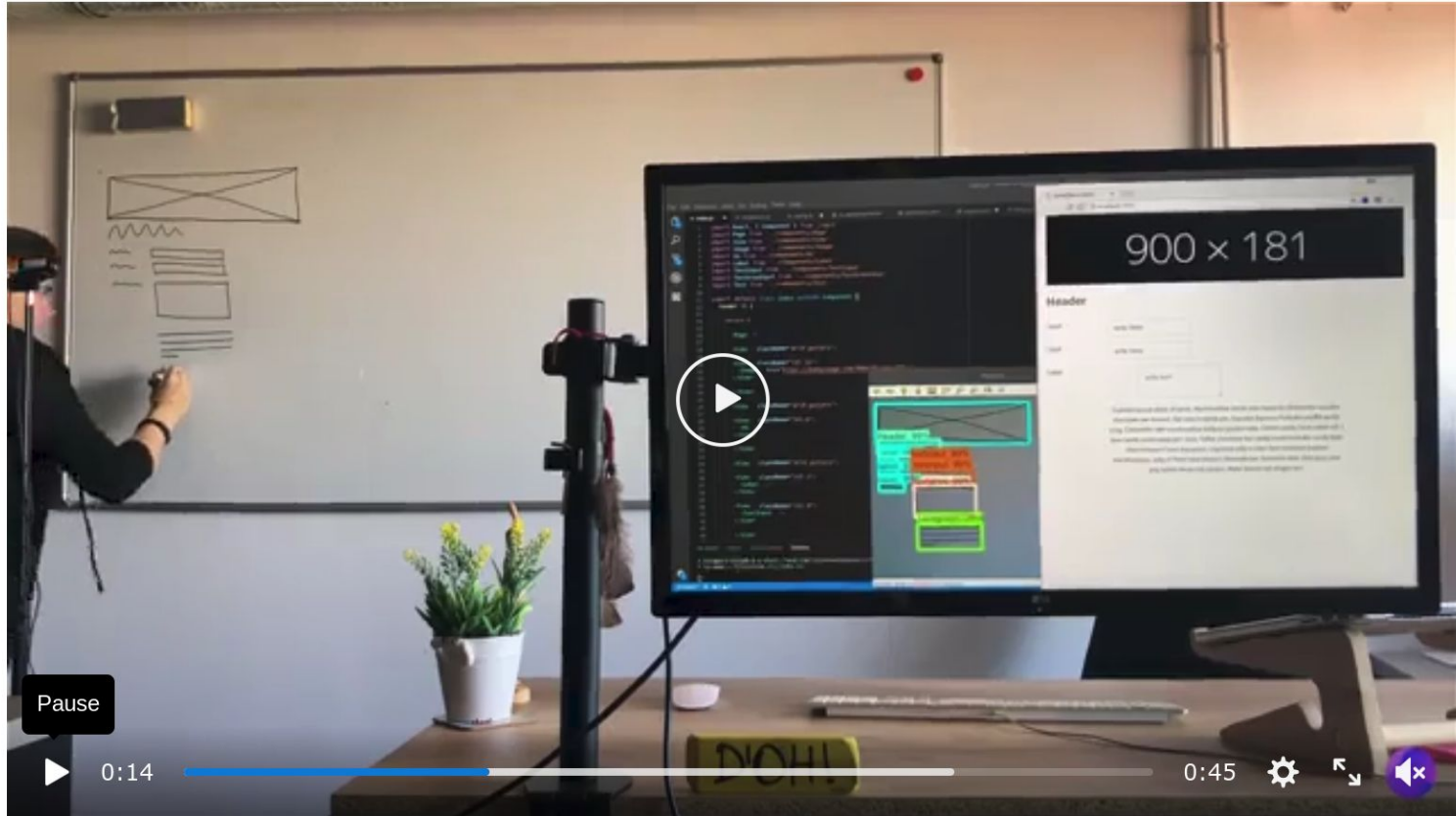
States

What about AI?

Posted by u/kshitijmishra23 2 days ago

1.4k

From wireframe to HTML Code. A great milestone achieved by AI for the frontend.



123 Comments Share Save ...

94% Upvoted

Summary

- Use sequence, not flow diagram
- Baseline the WSD on resources
- Add Transitions (hypermedia) to navigate resources
- Add Parameters to modify state
- Generate other assets:
 - Machine Formats
(Schema, OAI, RAML, RDFS, ALPS, DCAP, etc.)
 - Human Formats (reference, documentation)
 - Server Scaffolding (code)

Using Web Sequence Diagrams with your APIs

Mike Amundsen
API Academy
@mamund

