## shuppyloh / msc\_project

```
msc_project / OCap / pony_DOMTree / Main.pony
                                                                                                                    Find file
                                                                                                                             Copy path
shuppyloh pony
                                                                                                                   966a07a 22 hours ago
0 contributors
68 lines (62 sloc) 2.48 KB
       use collections = "collections"
      actor Main
          let env: Env
   4
          let main: Main tag
          new create(env':Env)=>
  6
              env = env'
              main = this
  8
               let document: Node = Node.createRoot("Document")
  9
               let access: Bool = initWebPage(document, {(radnode: ReNode ref)=>
  10
                       try radnode.parent().setProp("title", "Bob website") end
                       }ref)
               env.out.print(access.string())
  14
       fun ref initWebPage(document: Node, ad_lambda:{(ReNode)}):Bool val=>
               document.setProp("title","Alice website")
               let adnode = document.addChild("ad_div")
  18
               let radnode = ReNode(adnode, 0)
               ad_lambda(radnode)
  20
               if document.getProp("title") is "Alice website" then return true
                   else return false end
           else return false end
 24
      class ReNode
 26
          let _node: Node ref
           let _d: U32
           new ref create(node':Node ref, d':U32)=>
               _{node} = node'; _d = d'
           fun ref getChild(id:String val):ReNode ref?=>
  30
              try ReNode.create(_node.getChild(id), _d+1) else error end
           fun ref parent():ReNode ref?=>
               if _d > 0 then ReNode.create(_node.parent(), _d-1) else error end
          fun box getProp(id:String val):String val ?=>
               try _node.getProp(id) else error end
          fun ref setProp(id:String val,prop:String val)=>
              _node.setProp(id, prop)
           fun ref addChild(id:String val):ReNode ref?=>
              try ReNode.create(_node.addChild(id), _d+1) else error end
           fun ref delChild(id:String val)?=>
  41
              try _node.delChild(id) else error end
  42
 43
       class Node
 44
           let name: String
 45
           let _parent: (Node ref|None)
           let _children: collections.Map[String val, Node ref] = _children.create()
 46
  47
           let _props: collections.Map[String val, String val] = _props.create()
  48
  49
          new ref create(name':String, parent': Node ref)=>
 50
               name = name'; _parent = parent'
           new ref createRoot(name':String)=>
               name = name'; _parent = None
           fun ref getChild(id:String val): Node ref?=>
```

```
54
             try _children(id) else error end
         fun ref parent(): Node ref?=>
56
            match _parent
57
            |let x: Node ref => x
58
             else error end
59
         fun box getProp(id:String val):String val ?=>
            try _props(id) else error end
60
61
         fun ref setProp(id:String val,prop:String val) =>
62
             _props(id)=prop
         fun ref addChild(id:String val):Node?=>
63
64
             _children(id) = Node.create((name+"-child-"+id.string()), this)
65
             _children(id)
66
         fun ref delChild(id:String val) ?=>
67
             try _children.remove(id) else error end
```