

Dec 05, 05 13:11

**exception.scm**

Page 1/1

```
(define (with-handler handler proc)
  (call-with-current-continuation
   (lambda (esc)
     (proc (lambda (reason)
            (call-with-current-continuation
             (lambda (restart)
               (esc (handler reason restart))))))))))

(with-handler
 (lambda (reason restart)
  (if (eq? reason 'failure)
      (begin
        (display "Something went wrong")
        (newline))
      (restart #f)))
 (lambda (throw)
  (display "1") (newline)
  (display "2") (newline)
  (throw 'annoyed)
  (display "3") (newline)
  (display "4") (newline)
  (throw 'failure)
  (display "5") (newline)
  (display "6") (newline)))
```

Dec 05, 05 13:18

**wishlist.scm**

Page

```
(define-structure surklet surklet-interface
  (open surklets
    scheme-with-scsch)
  (begin

    (define (ask-for-wish)
      (let ((wish-input (make-text-field))
            (submit-button (make-submit-button)))
        (let ((req
              (send-html/suspend
               (lambda (k-url)
                 `(html
                  (body
                   (h1 "Enter your wish")
                   (surklet-form ,k-url
                                (p "Your wish:"
                                   ,wish-input
                                   ,submit-button)))))))
          (input-field-value wish-input
                             (get-bindings req))))))

    (define (print-wish wish)
      (send-html/finish
       `(html
        (body
         (h1 "Your (short) wishlist" )
         (p ,wish)))))

    (define (main req)
      (print-wish (ask-for-wish))))
```

Dec 05, 05 13:46

wishlist-pro.scm

Page 1/2

```

(define-structure surflet surflet-interface
  (open scheme-with-scsch
    surflets
    (subset srfi-1 (filter))
    inspect-exception)
  (begin
    (define *wish-id* 0)

    (define (fresh-wish-id)
      (set! *wish-id* (+ *wish-id* 1))
      *wish-id*)

    (define (make-wish text)
      (cons (fresh-wish-id) text))

    (define wish-id car)
    (define wish-text cdr)

    (define (output-wish-list wish-list delete-button k-url)
      `(ul
        ,@(map
          (lambda (wish)
            `(li ,(wish-text wish) " "
              (url ,(delete-button k-url (wish-id wish))
                "delete wish"))
            wish-list)))

    (define (manage-wish-list wish-list)
      (let ((wish-input (make-text-field))
            (no-more-wishes (make-address))
            (submit-button (make-submit-button))
            (delete-button (make-annotated-address)))

        (let ((req
              (send-html/suspend
                (lambda (k-url)
                  `(html
                    (body
                     (h1 "Wishlist Manager Pro")
                     (surflet-form
                      ,k-url POST
                      ,(output-wish-list wish-list delete-button k-url)
                      (p "Add wish:" ,wish-input ,submit-button))
                     (p (url ,(no-more-wishes k-url)
                          "Perfectly happy"))))))))

          (case-returned-via (get-bindings req)
            ((no-more-wishes) wish-list)
            ((delete-button) => (lambda (id)
              (manage-wish-list
                (filter (lambda (wish)
                  (not (= id (wish-id wish))))
                  wish-list))))

            (else
             (manage-wish-list
              (cons (make-wish
                (input-field-value wish-input (get-bindings req)))
                wish-list))))))

    (define (main req)
      (with-inspecting-handler

```

Dec 05, 05 13:46

wishlist-pro.scm

Page

```

      8888
      (lambda (condition)
        (with-current-output-port (current-error-port)
          (display "starting remote handler for condition ")
          (display condition)
          (newline)
          (display "Please connect to port 8888")
          (newline)))
      (lambda ()
        (send-html/finish
         `(html
          (body
           (h2 "Your wishlist")
           (ul
            ,@(map (lambda (wish) `(li ,(wish-text wish)))
              (manage-wish-list '()))))))))
    ))

```