



わかった気になる

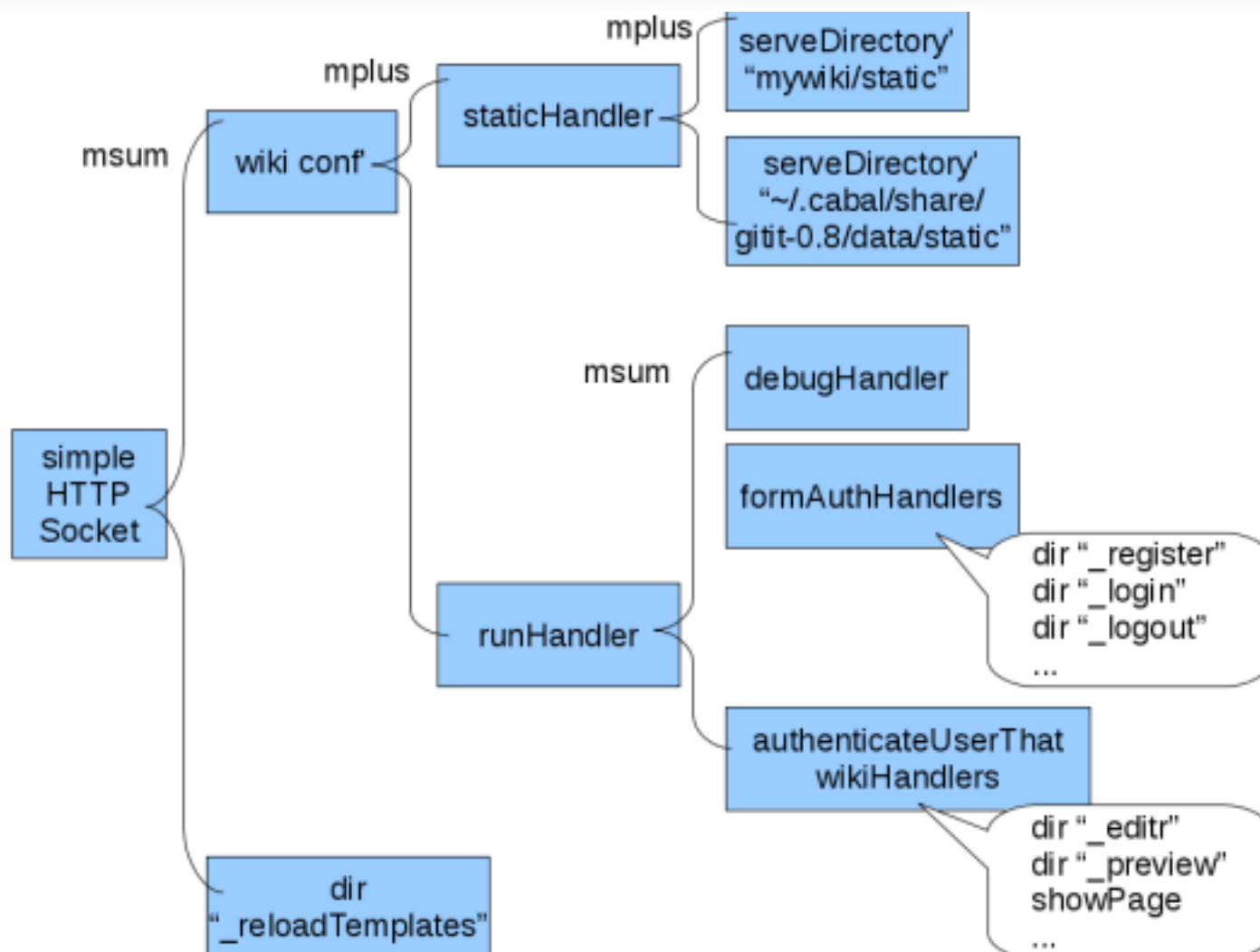
gitit-0.8

初心者Haskell勉強会

Kiwamu Okabe



ルーティング





monadplus

handlerを**mplus**で結合してるだけに見えます。失敗は**mzero**になるんですが、成功した場合には次のhandlerを実行してほしいように見えるのですが、、、そーゆーもの???



ページ表示

```
indexPath :: Handler
indexPath = do
  path' <- getPath
  base' <- getWikiBase
  let prefix' = if null path' then "" else path' ++ "/"
  fs <- getFileStore
  listing <- liftIO $ directory fs prefix'
  let isDiscussionPage (FSFile f) = isDiscussPageFile f
      isDiscussionPage (FSDirectory _) = False
  let prunedListing = filter (not . isDiscussionPage) listing
  let htmlIndex = fileListToHtml base' prefix' prunedListing
  formattedPage defaultPageLayout{
    pgPageName = prefix',
    pgShowPageTools = False,
    pgTabs = [],
    pgScripts = [],
    pgTitle = "Contents"} htmlIndex
```





gitアクセス

getFileStoreでFileStoreを呼出^(*1)

```
randomPage :: Handler
randomPage = do
  fs <- getFileStore
  files <- liftIO $ index fs
  let pages = map dropExtension $
                filter (\f -> isPageFile f && not (isDiscussPageFile f)) files
  base' <- getWikiBase
  if null pages
    then error "No pages found!"
```

(*1) <http://hackage.haskell.org/packages/archive/filestore/latest/doc/html/Data-FileStore-Types.html>





認証

```
authenticateUserThat predicate level handler = do
  cfg <- getConfig
  if level <= requireAuthentication cfg
  then do
    mbUser <- getLoggedInUser
    rq <- askRq
    let url = rqUri rq ++ rqQuery rq
    case mbUser of
      Nothing    -> tempRedirect ("/_login?" ++
                                   urlEncodeVars [("__destination", url)]) $ toResponse ()
      Just u     -> if predicate u
                    then handler
                    else error "Not authorized."
  else handler
```





gititプラグイン#1

```
gitit/Network/Gitit/Plugins.hs
```

がプラグイン読み込みコード。^(*2)

^(*2)詳細は <http://www.bluishcoder.co.nz/2008/11/dynamic-compilation-and-loading-of.html> を参照





gititプラグイン#2

```
loadPlugin :: FilePath -> IO Plugin
loadPlugin pluginName = do
  --snip--
  pr <- findModule (mkModuleName "Prelude") Nothing
  i <- findModule (mkModuleName "Network.Gitit.Interface") Nothing
  m <- findModule (mkModuleName modName) Nothing
  setContext []
  #if MIN_VERSION_ghc(7,0,0)
    [(m, Nothing), (i, Nothing), (pr, Nothing)]
  #else
    [m, i, pr]
  #endif
  value <- compileExpr (modName ++ ".plugin :: Plugin")
  let value' = (unsafeCoerce value) :: Plugin
  return value'
```





ghc7だと動かない

```
$ ghc --version
The Glorious Glasgow Haskell Compilation System, version 6.12.1
$ gitit -f default.conf
Loading plugin 'plugins/TwitterUrl.hs'...
Finished loading plugins.
$ ghc --version
The Glorious Glasgow Haskell Compilation System, version 7.0.3
$ gitit -f default.conf
Loading plugin 'plugins/TwitterUrl.hs'...
gitit: This ELF file contains no symtab
gitit: gitit: panic! (the 'impossible' happened)
      (GHC version 7.0.3 for x86_64-unknown-linux):
          loadArchive "/usr/lib/ghc-7.0.3/ghc-7.0.3/libHSghc-7.0.3.a": failed

Please report this as a GHC bug:  http://www.haskell.org/ghc/reportabug
```



plugins 良いらしい



@shelarcy

shelarcy (しえらーしい)

最新版に誘導

<http://hackage.haskell.org/package/plugins>

RT @master_q: これはもう古いみたい

hs-plugins Dynamically Loaded Haskell
Modules <http://j.mp/ljxj6y>

1 Jun via Tween JA Favorite Retweet RT Reply



plugins使い方#1

API.hs

```
module API where

data Test = Test {
    field :: String
}

test :: Test
test = Test { field = "default value" }
```





plugins使い方#2

Test.hs

```
module Test where
```

```
import API
```

```
resource = test { field = "success" }
```



plugins使い方#3

Main.hs

```
import System.Plugins
import API

main = do
  m_v <- load_ "../Test.o" ["../api"] "resource"
  v <- case m_v of
    LoadFailure _ -> error "load failed"
    LoadSuccess _ v -> return v
  let s = field v
  print s -- => "success"と表示
```

