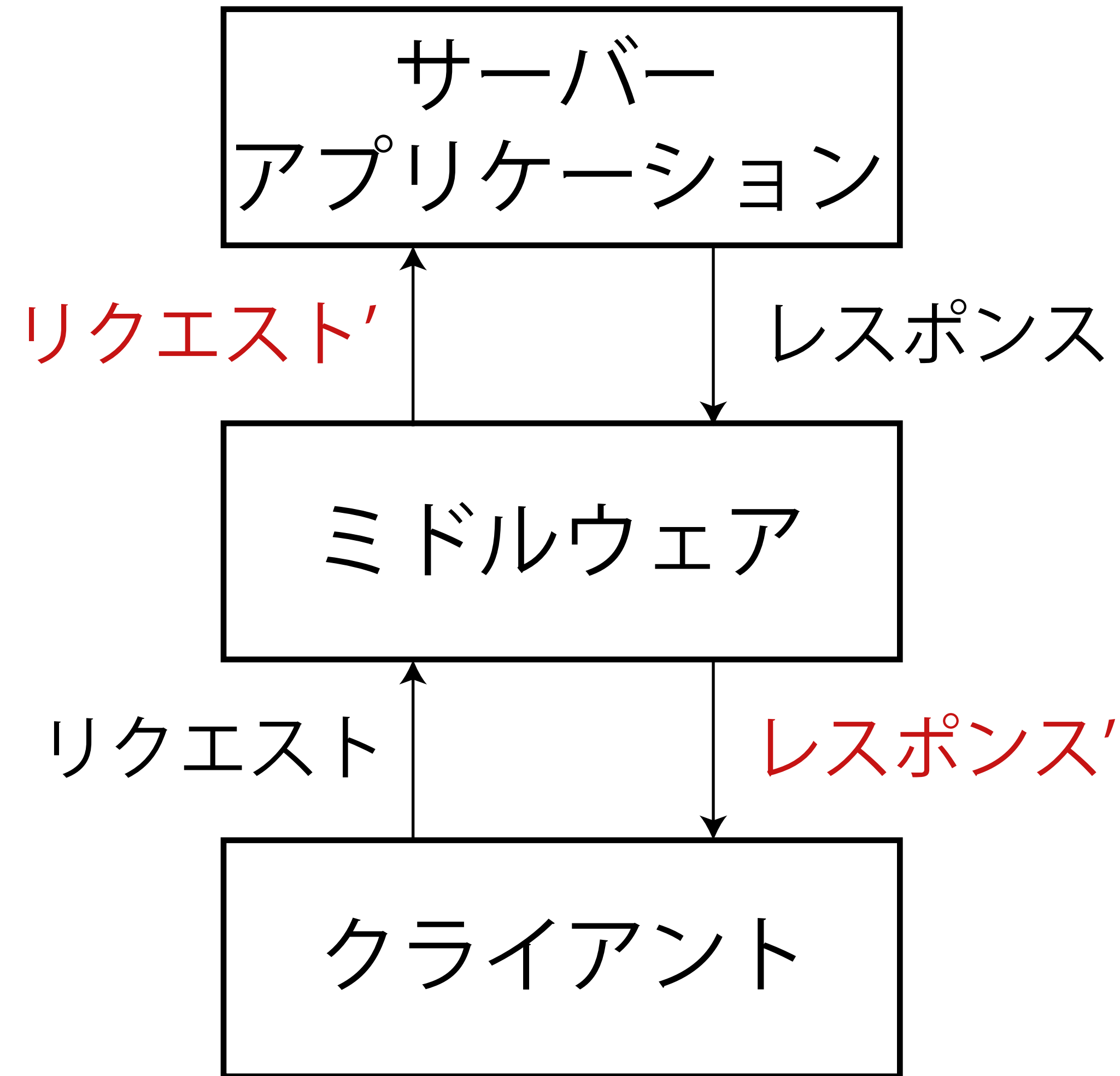


ミドルウェアの作り方

@fumieval

概要



アプリケーションに被せて、リクエストとレスポンスを変更できる

何ができる？

- 新しいルートを生やす
 - e.g. /assets/style.css のリクエストが来た時、静的ファイルとして返す
- メタデータを付与する
 - e.g. 認証する、リクエスト ID を付与する
- アクセスを制御する
 - e.g. 許可されていないページには 403 を返す
- ロギング

wai のアプリケーション

```
type Application
```

```
  = Request
```

```
  -> (Response -> IO ResponseReceived)
```

```
  -> IO ResponseReceived
```

```
app req sendResp = ...
```

wai のミドルウェア

```
type Middleware = Application -> Application
```

```
middleware app req sendResp  
  = app  
    (modifyRequest req)  
    (sendResp . modifyResponse)
```

```
naiveLogger :: Middleware
naiveLogger app req sendResp =
  app req $ \resp -> do
    putStrLn $ "Request: " <> show req
    putStrLn $ "Response: " <> show resp
    sendResp resp
```


vault

```
vault :: Request -> Vault
```

A location for arbitrary data to be shared by applications and middleware.

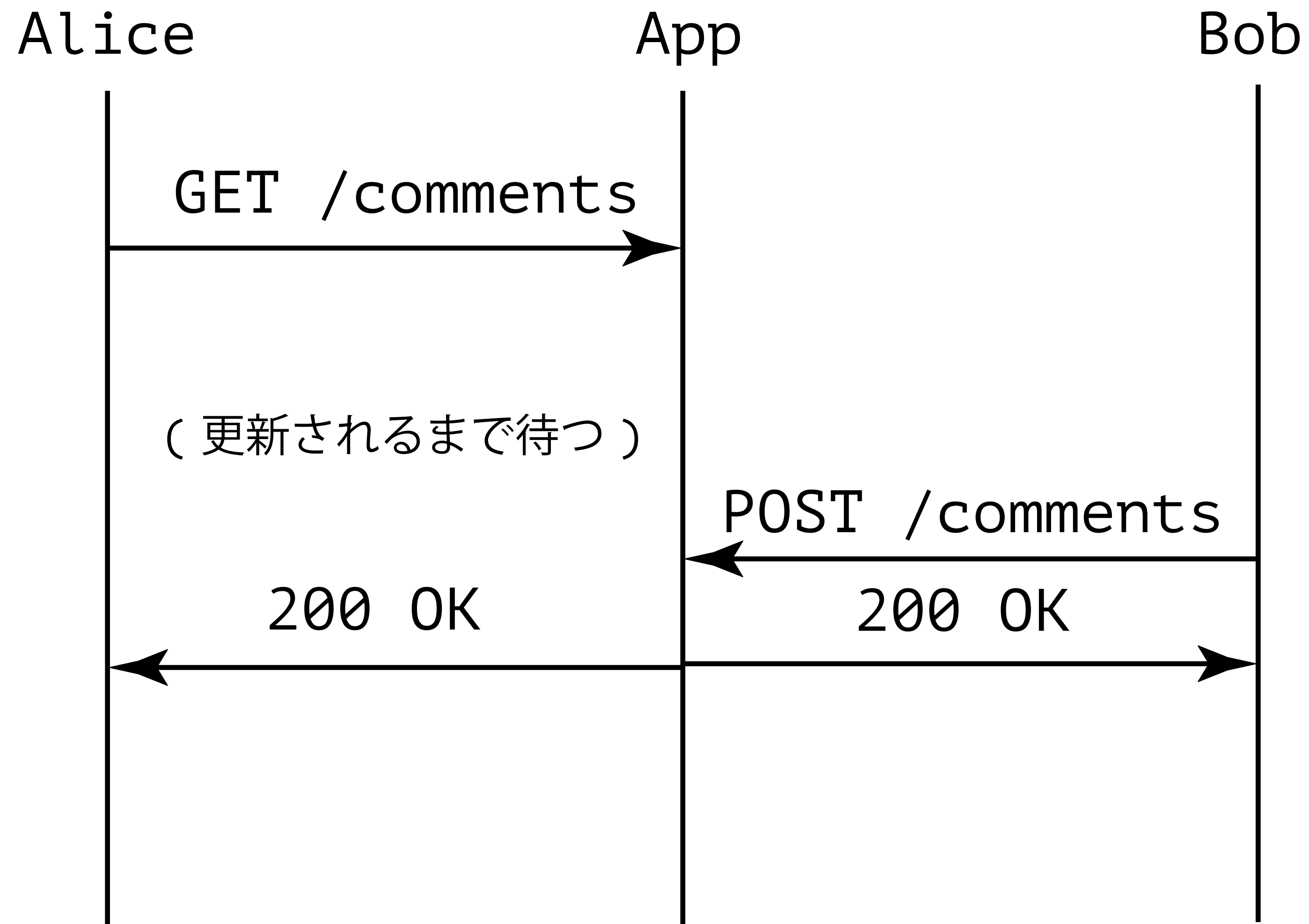
```
newKey :: IO (Key a)
```

```
lookup :: Key a -> Vault -> Maybe a
```

```
insert :: Key a -> a -> Vault -> Vault
```

ミドルウェアからアプリケーションに情報を渡すための仕組み
(例えば認証したユーザー名などを渡せる)

comet



ミドルウェアで実現する

- パス P に対し、
X-COMET-WAIT ヘッダーが指定されているなら
リクエストを保留する
- 同じパス P に対し、
POST, PUT, PATCH, DELETE のいずれかのリクエストが
成功 (2xx) したら
保留したリクエストを渡す

```
create :: IO Middleware
create = do
  comet <- new
  pure $ \app req sendResp -> do
    let key = pathInfo req
    case lookup "X-Comet-Wait" (requestHeaders req) of
      Just "true" -> wait comet key
      _ -> pure ()

  app req $ \resp -> do
    when (isImpureMethod (requestMethod req)
      && statusIsSuccessful (responseStatus resp))
      $ notify comet key
    sendResp resp
```

保留する

保留を解除する

```
type Comet k = TVar (Map k Int)
new :: IO (Comet k)
new = newTVarIO Map.empty

wait :: Ord k => Comet k -> k -> IO ()
wait v k = do
    original <- Map.findWithDefault 0 k <$> readTVarIO v
    atomically $ do
        current <- readTVar v
        when (current == original) retry

notify :: Ord k => Comet k -> k -> IO ()
notify = atomically $ modifyTVar' v $ M.insertWith (+) 1 k
```

(使い終わったキーを削除する場合もう少し複雑になる)

ミドルウェアいろいろ

wai-middleware-static	静的ファイルを配信する
wai-extra	たくさんミドルウェアがある
wai-middleware-prometheus	Prometheus のメトリクスを生成
wai-middleware-auth	OAuth や OIDC

