```
module Handler.Twert where

import Handler.Feed
import Import

postTwertR :: Handler Html
postTwertR = do
    ((result, widget), enctype) <- runFormPost $ twertForm
    case result of
        FormSuccess twert -> do
                             twertKey <- runDB $ insert twert
                             redirect FeedR
        _ -> defaultLayout
                [whamlet|
                    <p>Oops. Looks like there was an error. Please try agian.
                    <form method=post action=@{TwertR} enctype=#{enctype}>
                        <table>
                            ^{widget}
                        <button>Submit
                |]

getTwertR :: Handler Html
getTwertR = redirect FeedR



module Handler.Register where

import Import

personForm :: Form User
personForm = renderTable $ User
    <$> areq textField "Desired username" Nothing
    <*> areq emailField "Email" Nothing
    <*> areq passwordField "Password" Nothing

getRegisterR :: Handler Html
getRegisterR = do
    (widget, enctype) <- generateFormPost personForm
    defaultLayout
        [whamlet|
            <p>
                Enter your information to register!
            <form method=post action=@{RegisterR} enctype=#{enctype}>
                <table>
                    ^{widget}
                <button>Submit
        |]

postRegisterR :: Handler Html
postRegisterR = do
    ((result, widget), enctype) <- runFormPostNoToken personForm
    case result of
        FormSuccess person -> do
                              existing <- (\(User name _ _) -> runDB $ selectList [UserIdent ==. name] []) person
                              case null existing of
                                  True -> do
                                              personKey <- runDB $ insert $ person
                                              defaultLayout [whamlet|Success! Here is your person: #{show personKey}|]
                                  _    -> do
                                              defaultLayout
                                                  [whamlet|
                                                      <p>That username is already taken. Please try another.
                                                      <form method=post action=@{RegisterR} enctype=#{enctype}>
                                                          <table>
                                                              ^{widget}
                                                          <button>Submit
                                                  |]
        _ -> defaultLayout
                [whamlet|
                    <p>Invalid input. Let's try again.
                    <form method=post action=@{RegisterR} enctype=#{enctype}>
                        <table>
                            ^{widget}
                        <button>Submit
                |]



module Handler.GetTwert where
```

```
import Import

getGetTwertR :: TwertId -> Handler Html
getGetTwertR twert = do
    twerts <- runDB $ selectList [TwertId ==. twert] []
    case not $  null twerts of
        True -> do
                    defaultLayout
                        [whamlet|
                                <h1>Twert:
                                $forall Entity twertId twert <- twerts
                                    <p><b>Title:</b> #{twertTitle twert}
                                    <p><b>Content:</b> #{twertContent twert}
                                    <p>Id: #{show twertId}
                                    <br>
                        |]
        _    -> do
                    defaultLayout
                        [whamlet|
                            <p>No twert found
                        |]


module Handler.Feed where

import Import

twertForm :: Html -> MForm Handler (FormResult Twert, Widget)
twertForm = renderTable $ Twert
    <$> areq textField "Title:" Nothing
    <*> areq textField "Content:" Nothing
    <*> lift (liftIO getCurrentTime)
    -- Let's just give them the user id of 1 for testing purposes.
    <*> areq hiddenField "" (Just 1)

-- Here we basically authenticate everyone who looks here.
ggetAuth :: (IsString a, Monad m) => m (Maybe a)
ggetAuth = do return $ Just "hi"

getFeedR :: Handler Html
getFeedR = do
    (widget, enctype) <- generateFormPost twertForm
    mAuthId <- ggetAuth
    twerts <- runDB $ selectList ([] :: [Filter Twert]) [Desc TwertTime]
    defaultLayout $ do
        setTitle "Twert Feed"
        $(widgetFile "twert-feed")
```