

Yesod

Ein Haskell Webframework

***Eine kurze
Geschichte der ...***

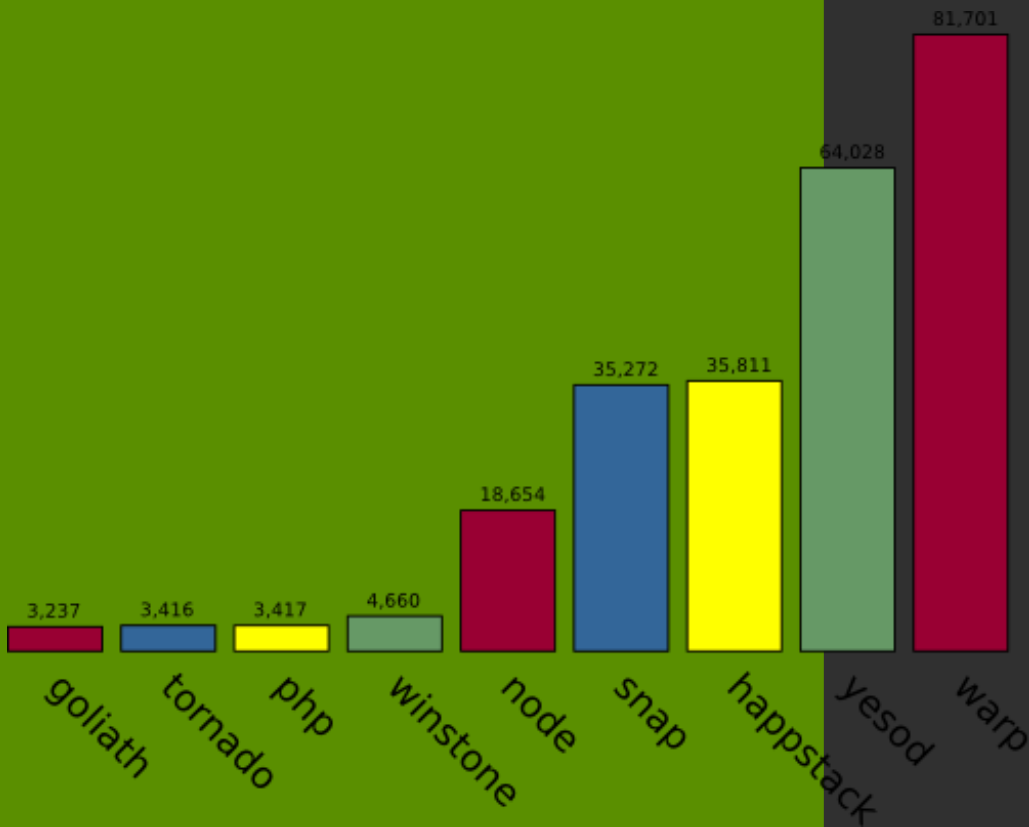
Next

- *Haskell ;-)*
- *Typsicherheit*
- *Geschwindigkeit*

Previous

Next

*Pong benchmark,
extra large instance,
requests/second*



[Quelle](#)

- *Shakespeare Templates ;)*
- *Widgets*
- *Forms*
- *Routing*
- *Authentication/Authorization*
- *Sessions*
- *Persistenz*
- *I18n*

Previous

Next

```
<div id=navigation>
  $maybe prev <-
prevSlide
  <a
href=@{SlideR (fst
prev) }>
    <button
class="button
green">Previous
  $nothing
  $maybe next <-
nextSlide
  <a
href=@{SlideR (fst
next) }>
    <button
class="button
green">Next
  $nothing
```

[Previous](#)[Next](#)

```
#myid
  color: #{red}
  font-size:
    #{bodyFontSize}
  foo bar baz
  background-
    image:
      url (@{MyBackgroundR}
    )

section.blog {
  padding: 1em;
  border: 1px
    solid #000;
  h1 {
    color:
      #{headingColor};
  }
}
```

[Previous](#)[Next](#)

```
$(function() {  
  
  $(".section.#{section  
    Class}").hide();  
  
  $("#mybutton").click  
    (function() {  
  
    document.location =  
      "@{SomeRouteR}";  
      })  
      ^{addBling}  
    });  
});
```

[Previous](#)[Next](#)

- *Kleine Teile einer App (footer, header etc)*
- *Für jedes dieser Teile gibt es:*
 - *Html Code*
 - *Etwas Css*
 - *JavaScript*
- *Einige davon kommen in Header, andere in den Body*
- *Einschränkungen wie:*
 - Nur 1 Title darf existieren werden geprüft*
- *Widgets beachten diese Einschränkungen*
- *Folgen der richtigen Logik beim Zusammensetzen mehrerer Widgets*

- *Prüft Daten auf Validität*
- *Umwandlung in die geforderten Haskell Typen*
- *Generiert HTML Code für das Form*
- *Generiert JS für clientseitige Validation und Datepicker etc*
- *Forms können ineinander verschachtelt werden*

Previous

Next

```
/slide/new/#PresentationId NewSlideR  
POST  
/slide/#SlideId  
SlideR GET  
/slide/edit/#SlideId  
EditSlideR GET POST  
/slide/delete/#Slide  
Id DelSlideR GET
```

- *Typsicherheit*
- *Routen dürfen nicht überlappen*
- *Handler bearbeiten die Routen (Siehe später)*

[Previous](#)[Next](#)

- *First class support für PostgreSQL, SQLite and MongoDB.*
- *Automatische Schema Migrationen*
- *Wandelt Datenbanktypen in die richtigen Haskelltypen um*

Previous

Next

- *OpenID, BrowserID and OAuth*
- *Eigene Mechanismen leicht zu implementieren*

Previous

Next

- *Schnelles Setup von Seiten mit guter Qualität*
- *Config file parsing*
- *Effizientes Ausliefern statischer Dateien*
- *Gute Verzeichnisstruktur*

Previous

Next

- *Noch fehlender Tool support*
- *Haskell ist in der Wirtschaft noch nicht etabliert*

Previous

Next

***Jetzt
schauen wir
mal wie das
so aussieht
:-)***

Previous