

react-rails

React.js meetup #1



@hokaccha

reactjs / react-rails

Watch 56

★ Star 1,162

🍴 Fork 167

Ruby gem for automatically transforming JSX and using React in Rails.

📄 259 commits

🌿 6 branches

📦 13 releases

👤 41 contributors

🔄 branch: master react-rails / +

Merge pull request #248 from gbanis/patch-2

 rmosolgo authored 2 days ago

latest commit 0faa0f6f86

| | | |
|-----------------------|--|--------------|
| 📁 gemfiles | Test against Rails 4.2 | 3 months ago |
| 📁 lib | bump to 1.0 | 16 days ago |
| 📁 test | feat(ReactRailsUJS) mount/unmount components at node | a month ago |
| 📄 .gitignore | support new replay_console option that allows captured console.* mess... | a month ago |
| 📄 .travis.yml | Bundler is installed by default on Travis CI | 3 months ago |
| 📄 Appraisals | Test against Rails 4.2 | 3 months ago |
| 📄 Gemfile | Initial checkin | 2 years ago |
| 📄 LICENSE | Initial checkin | 2 years ago |
| 📄 README.md | Merge pull request #237 from christopherstyles/cs-small-proofreading-... | 2 days ago |
| 📄 Rakefile | - Enable emitting warnings by default | 6 months ago |
| 📄 VERSIONS.md | Fix a documentation typo and several small proofing edits | 16 days ago |
| 📄 react-rails.gemspec | Rspec tests fail due to `tilt` dependency | 7 days ago |

📖 README.md

Code

🔔 Issues 35

🔗 Pull requests 15

📊 Pulse

📈 Graphs

HTTPS clone URL

https://github.com/re

You can clone with HTTPS or Subversion. ?

🖥️ Clone in Desktop

📄 Download ZIP

Releases

Tags

Latest release

v1.0.0

ff564e8

v1.0.0

 rmosolgo released this 16 days ago · 8 commits to master since this release

react-rails 1.0.0

Let's call it 1.0! This release includes:

- JSX compilation in the Rails asset pipeline
- React component mounting with a view helper & UJS
- Support for different React builds in different environments
- Server-side rendering
- Component generator

Downloads

[Source code \(zip\)](#)[Source code \(tar.gz\)](#)

v0.13.0.0

Key Features

- JSX compilation in the Rails asset pipeline
- React component mounting with a view helper & UJS
- Server-side rendering

Mounting with view helper & UJS

MyComponent.jsx

```
class MyComponent extends React.Component {  
  render() {  
    return <div>Name: {this.props.name}</div>;  
  }  
}
```

view helper

```
<%= react_component('MyComponent', name: 'hokaccha') %>
```

```
# =>
```

```
<div
```

```
  data-react-class="MyComponent"
```

```
  data-react-props='{ "name": "hokaccha" } '>
```

```
</div>
```


react_ujs

```
// ※ コードはイメージ
```

```
findAll('[data-react-class]').forEach((el) => {  
  var reactClass = el.dataset.reactClass;  
  var reactProps = el.dataset.reactProps;  
  
  var component = window[reactClass];  
  var props = JSON.parse(reactProps);  
  
  var reactElement = React.createElement(component, props);  
  React.render(reactElement, el);  
});
```

- マウントするコードを書かなくていい
- APIリクエストしない分高速

Server Side Rendering

```
<%= react_component('MyComponent',  
    { name: 'hokaccha' }, { prerender: true }) %>
```

```
# =>
```

```
<div
```

```
  data-react-class="MyComponent"
```

```
  data-react-props='{"name": "hokaccha"}'>
```

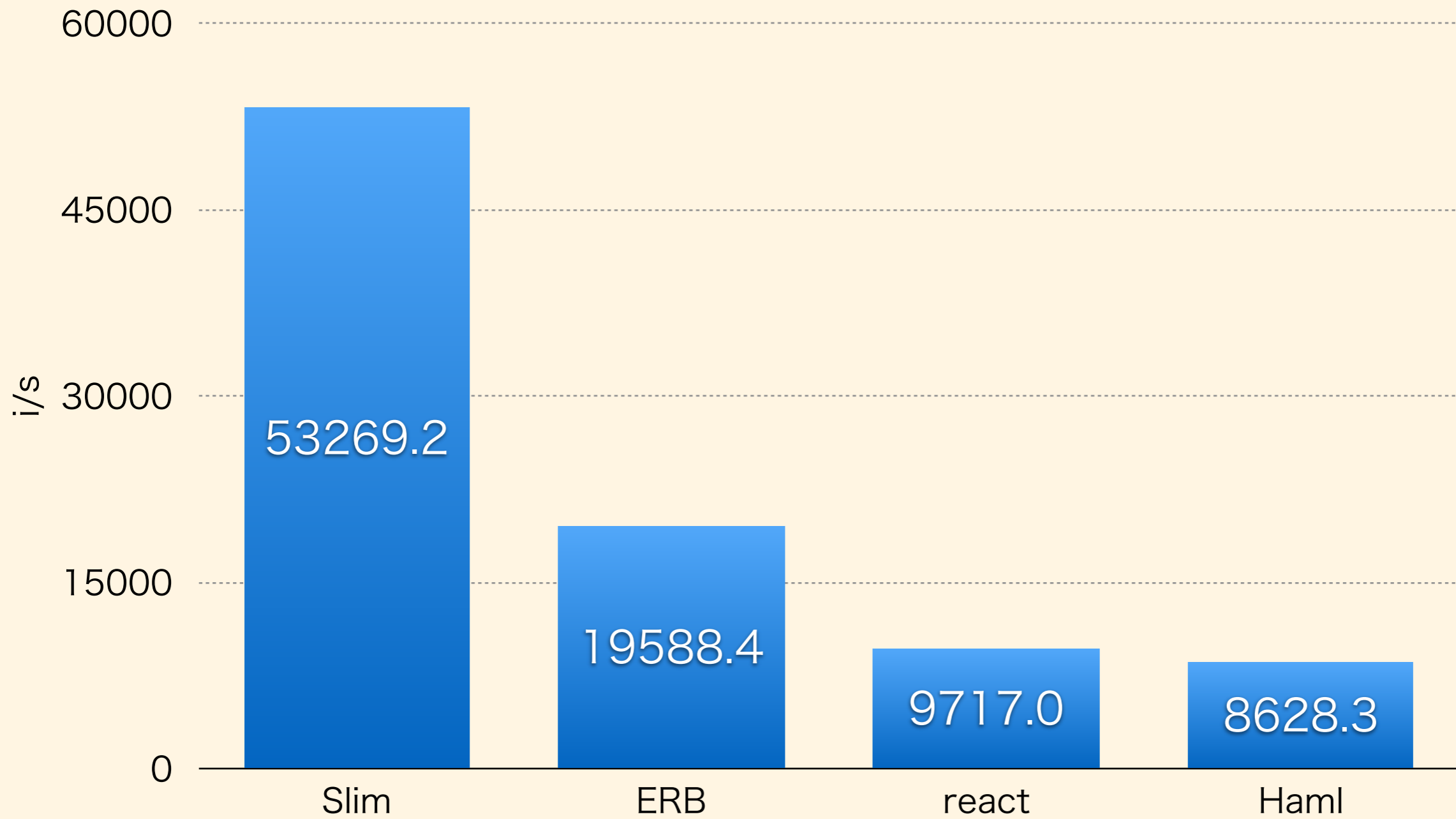
```
  <div data-react-checksum="xxx">Name: hokaccha</div>
```

```
</div>
```

- ExecJSでJSを実行
- therubyracer (V8) 推奨

DEMO

Performance



- 他のテンプレートエンジンと混ぜて使うので単純な比較はあまり意味が無い
- 死ぬほど遅いわけではなさそうというのがわかるためのベンチマーク
- Railsだと他の部分が遅いので全体のパフォーマンスとの割合が重要

Using with Turbolinks

```
<%= react_component('MyComponent',  
  { name: 'hokaccha' },  
  { prerender: !turboLinks_request? }) %>
```

- **サーバーサイドレンダリングが最初の
リクエスト時だけになる**
- **クライアントサイドでのルーティングが不要**
- **UJSがMount/Unmountの面倒を見てくれる**

DEMO

- 当然APIベースのほうがCoolだし速い
- 細かい遷移の制御はできない (たぶん)
- レールに乗っかることで簡単にできる
ようになるというのが重要

サーバーサイドコンテキスト
共有問題（時間があれば）

まとめ

- グリングリン動くSPAとかにはたぶん向かない
- 部分的に動的なコンテンツがあるようなアプリケーションだと使いどころあるかも
- Turbolinksにワンチャンある可能性

Thanks