From `rails new` to App Store

Building iOS and Android Apps with Rails 5 and Turbolinks

Who We Are

Geoff Buesing @gbuesing

Trevor Turk @trevorturk

What we built

- a Rails chat app
- ...with a native iOS app
- ...and a native Android app

github.com/gbuesing/turbochat

What we used

- Rails 5
- ActionCable
- Turbolinks
- Turbolinks-iOS adapter
- Turbolinks-Android adapter

What we didn't use

- PhoneGap
- Cordova
- React Native
- etc

Why choose Turbolinks native adapters?

- No new language, no new frameworks
- Minimal native adapters written in Swift and Java
- Simply use your existing web app for most native screens
- Break out of the web view box when needed

Overview of Turbolinks

Turbolinks makes your web application faster.

"Get the performance benefits of a single-page application without the added complexity of a client-side JavaScript framework."

github.com/turbolinks/turbolinks

Turbolinks history lesson (part one):

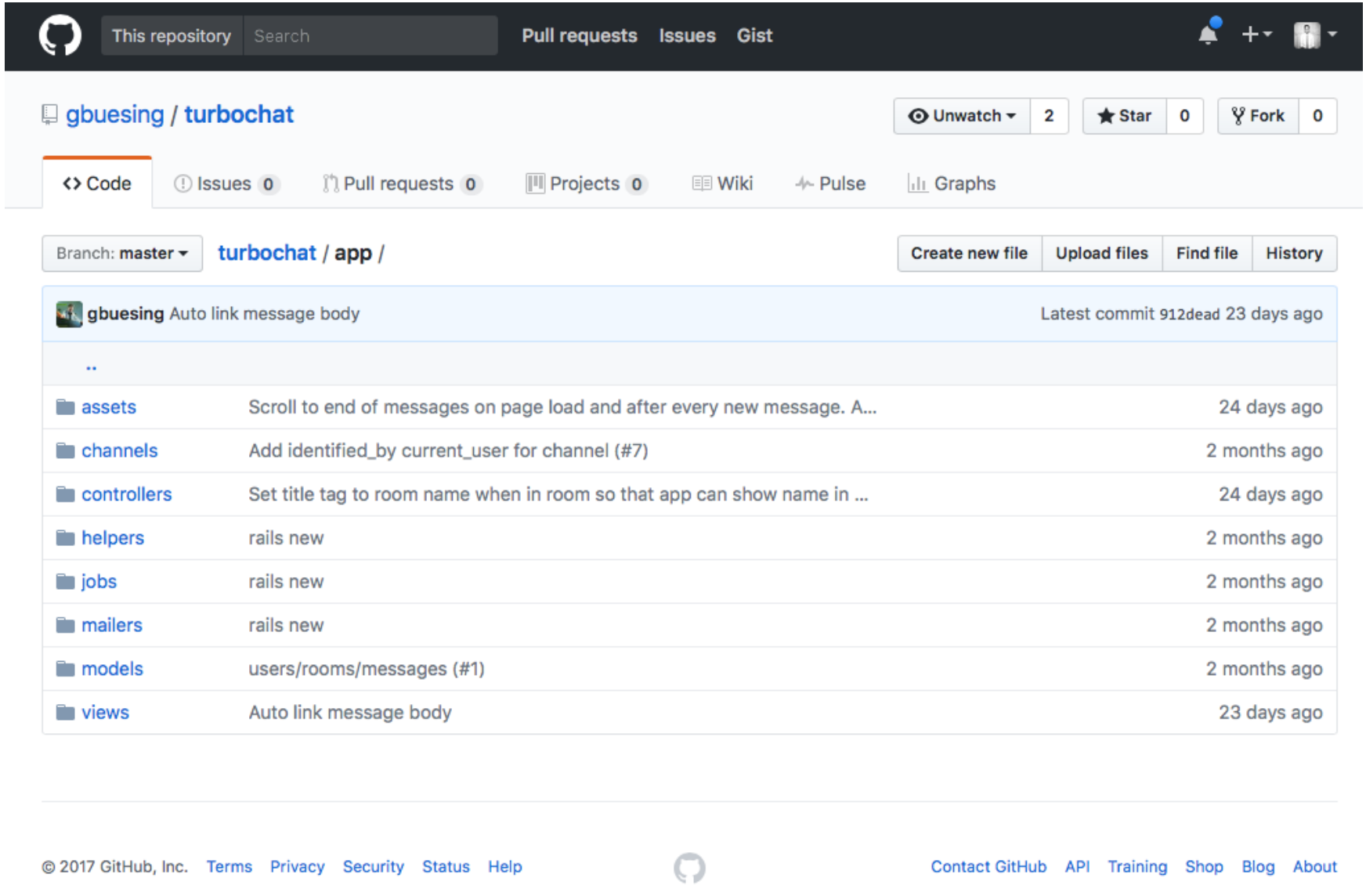
- pjax
- Turbolinks "Classic" (Rails 4.0)
- Turbolinks 5 (Rails 5.0)

pjax

Jquery plugin to make GitHub's inline code browser faster.

- Replace page elements without a full page reload
- Use jQuery and pushState() to maintain URLs

github.com/defunkt/jquery-pjax


This repository
Search
Pull requests
Issues
Gist
gbuesing / turbochat
Unwatch 2
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Pulse
Graphs
Branch: master
turbochat / app /
Create new file
Upload files
Find file
History
gbuesing Auto link message body
Latest commit 912dead 23 days ago
..
assets Scroll to end of messages on page load and after every new message. A... 24 days ago
channels Add identified_by current_user for channel (#7) 2 months ago
controllers Set title tag to room name when in room so that app can show name in ... 24 days ago
helpers rails new 2 months ago
jobs rails new 2 months ago
mailers rails new 2 months ago
models users/rooms/messages (#1) 2 months ago
views Auto link message body 23 days ago
© 2017 GitHub, Inc. Terms Privacy Security Status Help
Contact GitHub API Training Shop Blog About

```
<!DOCTYPE html>
<html>
<head>
  <script src="pjax.js"></script>
</head>
<body>
  <div id="file-navigation">
    <!-- links to folders, files, etc -->
  </div>
  <script>
    // enable faster browsing within the file-navigation div
    $(document).pjax('a', '#file-navigation')
  </script>
</body>
</html>
```

Turbolinks history lesson (part two):

- pjax
- Turbolinks "Classic" (Rails 4.0)
- Turbolinks 5 (Rails 5.0)

Turbolinks "Classic"

While pjax only replaces part of the page, Turbolinks operates on the entire page. When you follow a link, Turbolinks automatically fetches the page, swaps in its `<body>`, and merges its `<head>`, all without incurring the cost of a full page load."

github.com/turbolinks/turbolinks-classic

```
<!DOCTYPE html>
<html>
<head>
  <title>My application</title>
  <script src="turbolinks-classic.js"></script>
</head>
<body>
  <a href="/about">About us</a>
</body>
</html>
```

Turbolinks history lesson (part three):

- pjax
- Turbolinks "Classic" (Rails 4.0)
- Turbolinks 5 (Rails 5.0)

Turbolinks 5

- Ground-up rewrite with the same core idea as Turbolinks Classic
- Drops jQuery dependency
- Supports mobile apps via native adapters

github.com/turbolinks/turbolinks

Turbolinks adapters for iOS and Android

The native adapters automatically manage a single web view instance across multiple view controllers, giving you native navigation UI with all the client-side performance benefits of Turbolinks.

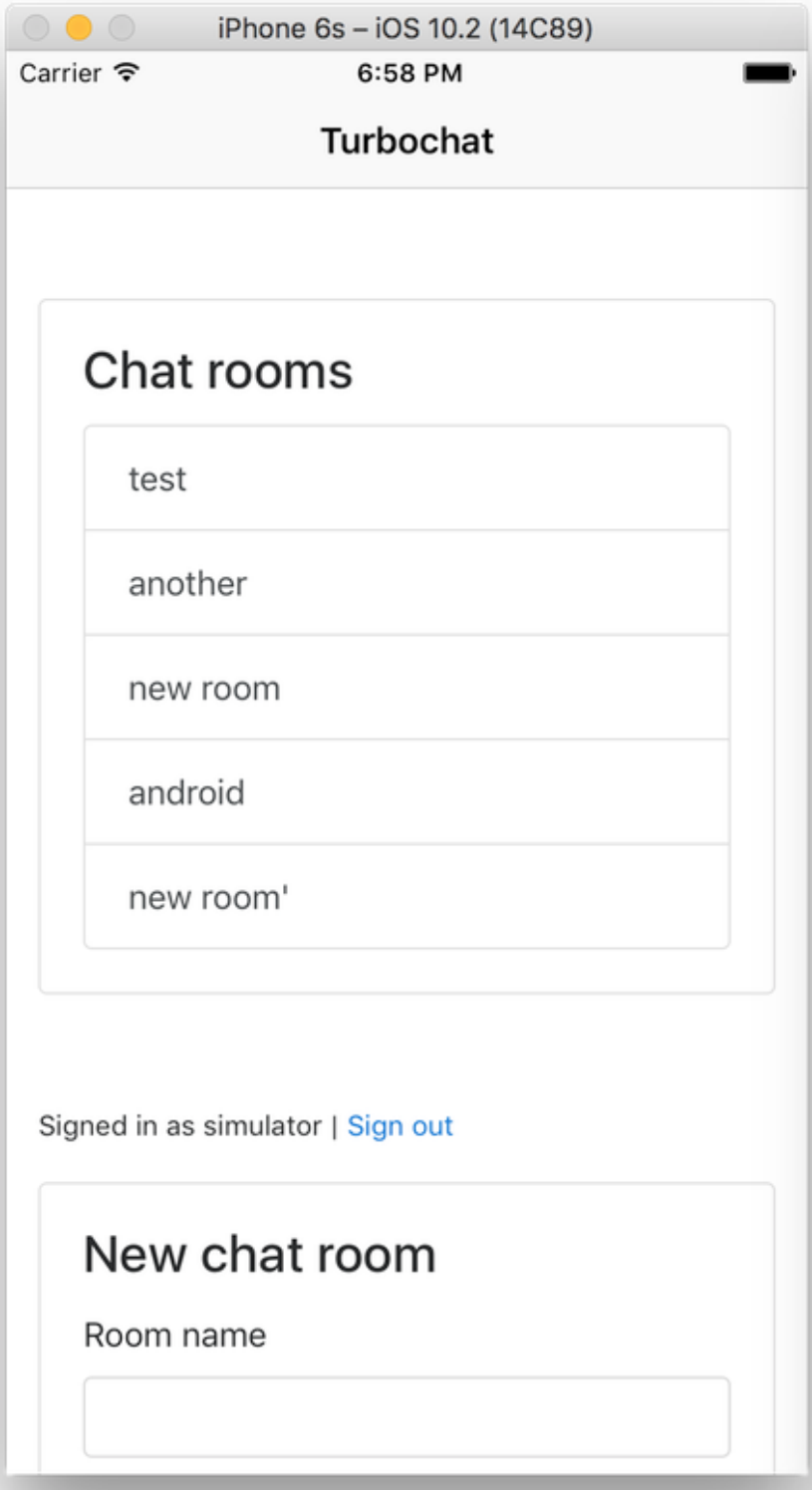
You can use native code and navigation UI progressively where needed. You're not locked into a custom app development framework.

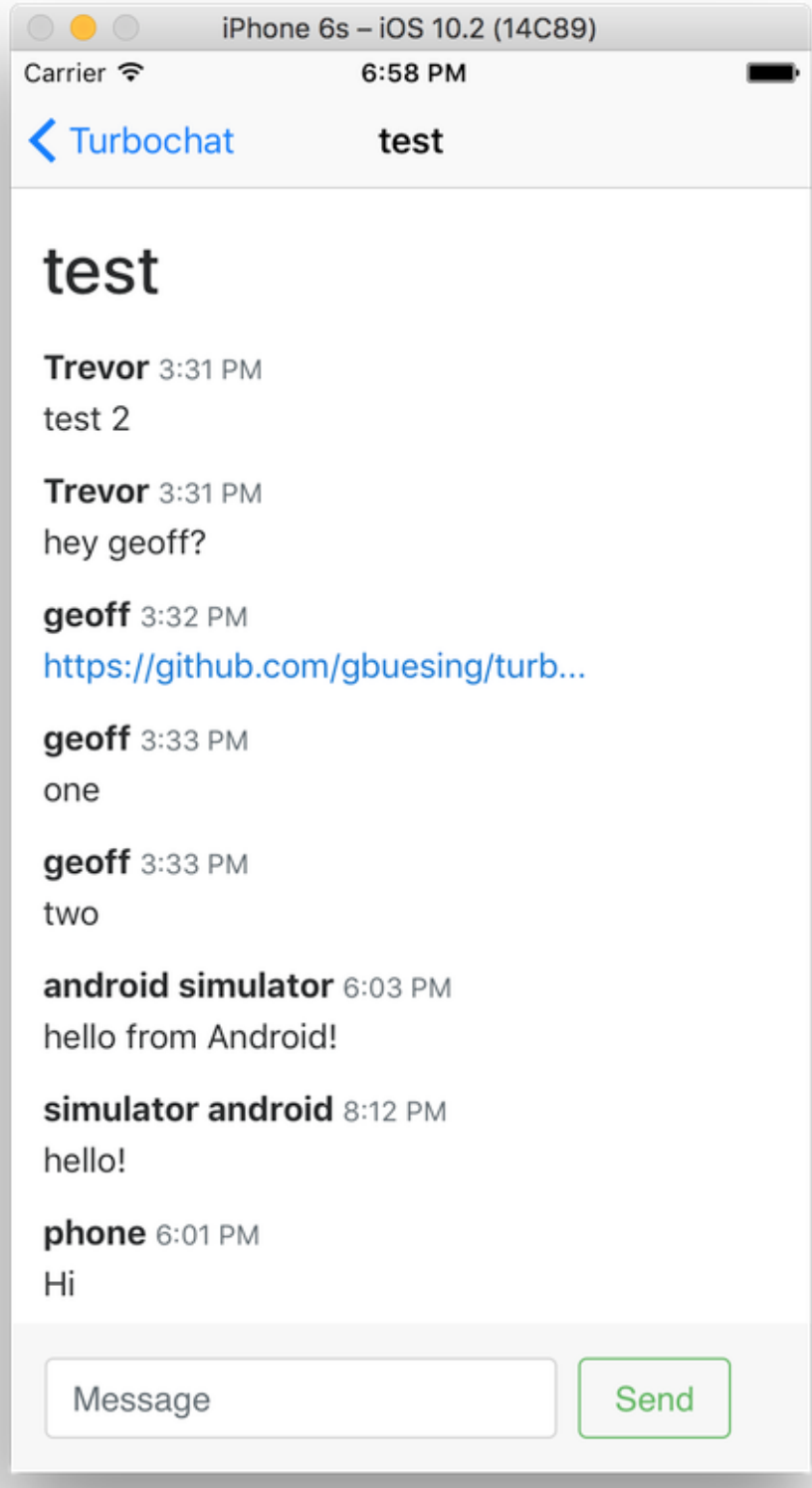
github.com/turbolinks/turbolinks-ios
github.com/turbolinks/turbolinks-android

Turbolinks adapter benefits:

- Leverage your existing web views on native platforms
- Use HTML/CSS for layouts, instead of complex storyboards etc
- Use the Safari developer mode and Chrome web tools you love
- Release native app bug fixes with a simple web deploy

The app: Turbochat


iPhone 6s – iOS 10.2 (14C89)
Carrier
6:58 PM
Turbochat
Chat rooms
test
another
new room
android
new room'
Signed in as simulator | Sign out
New chat room
Room name


iPhone 6s – iOS 10.2 (14C89)
Carrier
6:58 PM
Turbochat
test
test
Trevor 3:31 PM
test 2
Trevor 3:31 PM
hey geoff?
geoff 3:32 PM
https://github.com/gbuesing/turb...
geoff 3:33 PM
one
geoff 3:33 PM
two
android simulator 6:03 PM
hello from Android!
simulator android 8:12 PM
hello!
phone 6:01 PM
Hi
Message
Send

Breaking out of the web view box, i.e. hybrid apps

Communicating with your iOS and Android apps

- Web app can send messages to the native app via Javascript
- Native app can call Javascript functions in the web app

Example use cases

- Update web view cookie when device geolocation changes
- Display a native alert
- Add button to native toolbar

Web app sends message to iOS

```
// js
window.webkit.messageHandlers.log.postMessage("hi");
```

...iOS app exposes "log" handler:

```
// swift
webViewConfiguration.userContentController
  .addScriptMessageHandler(self, name: "log")

func userContentController(userContentController:
        WKUserContentController, didReceiveScriptMessage
    message: WKScriptMessage) {
    print("Logging: \(message.body)")
}
```

Web app sends message to Android

Use JavaScript to send a message from your web app:

```
// js
window.AndroidNative.log("hi")
```

...and the Android app recieves the message with data:

```
// java
@JavascriptInterface
public void log(String value) {
        Log.d("Logging:", value)
}

webView.addJavascriptInterface(this, "AndroidNative");
```

iOS calls web app function

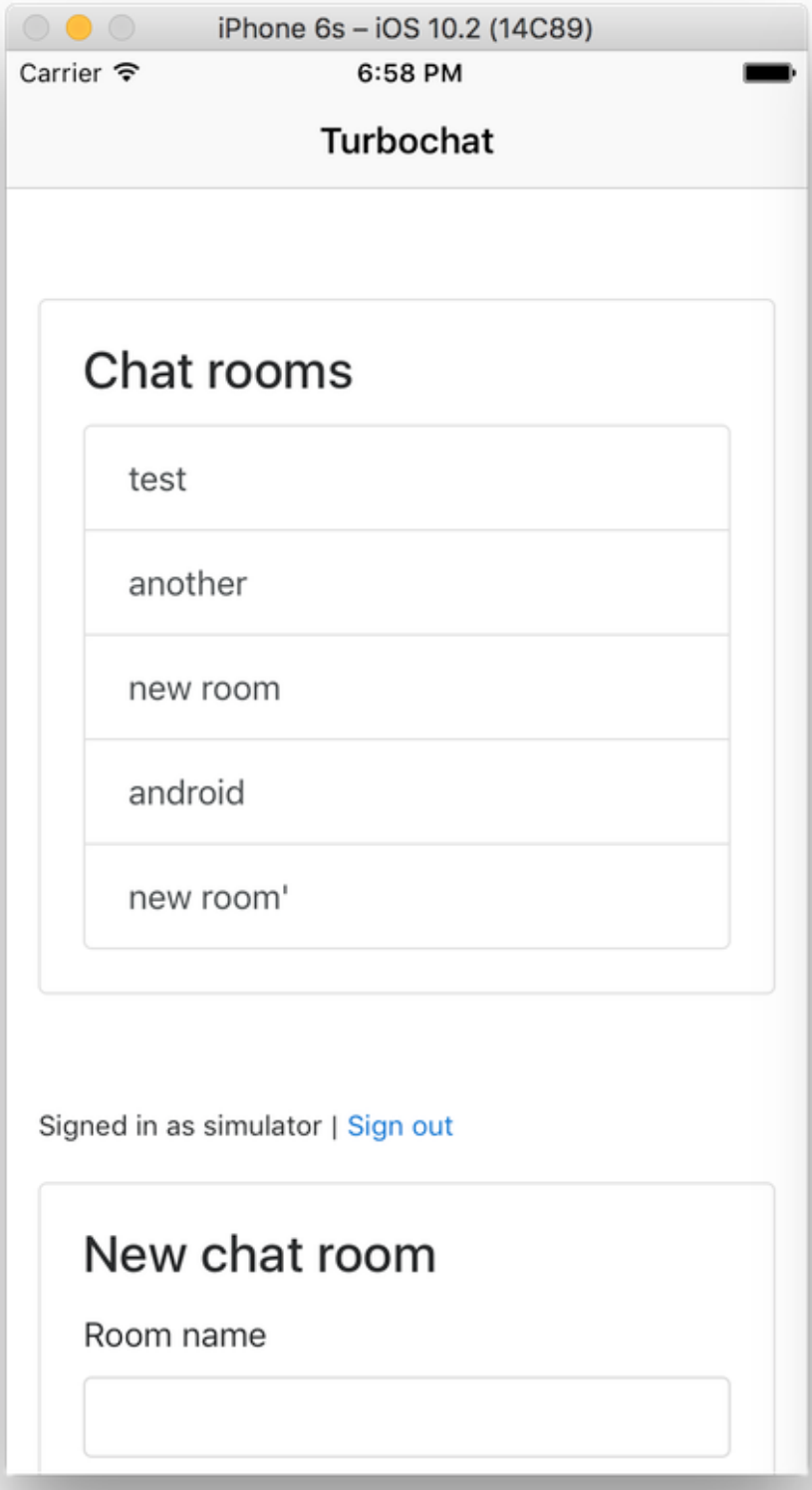
```swift
// swift
let js = "(function() { return new Date() })();";

webView.evaluateJavaScript(js) { (result, error) in
  print("Date is \(result)")
}
```

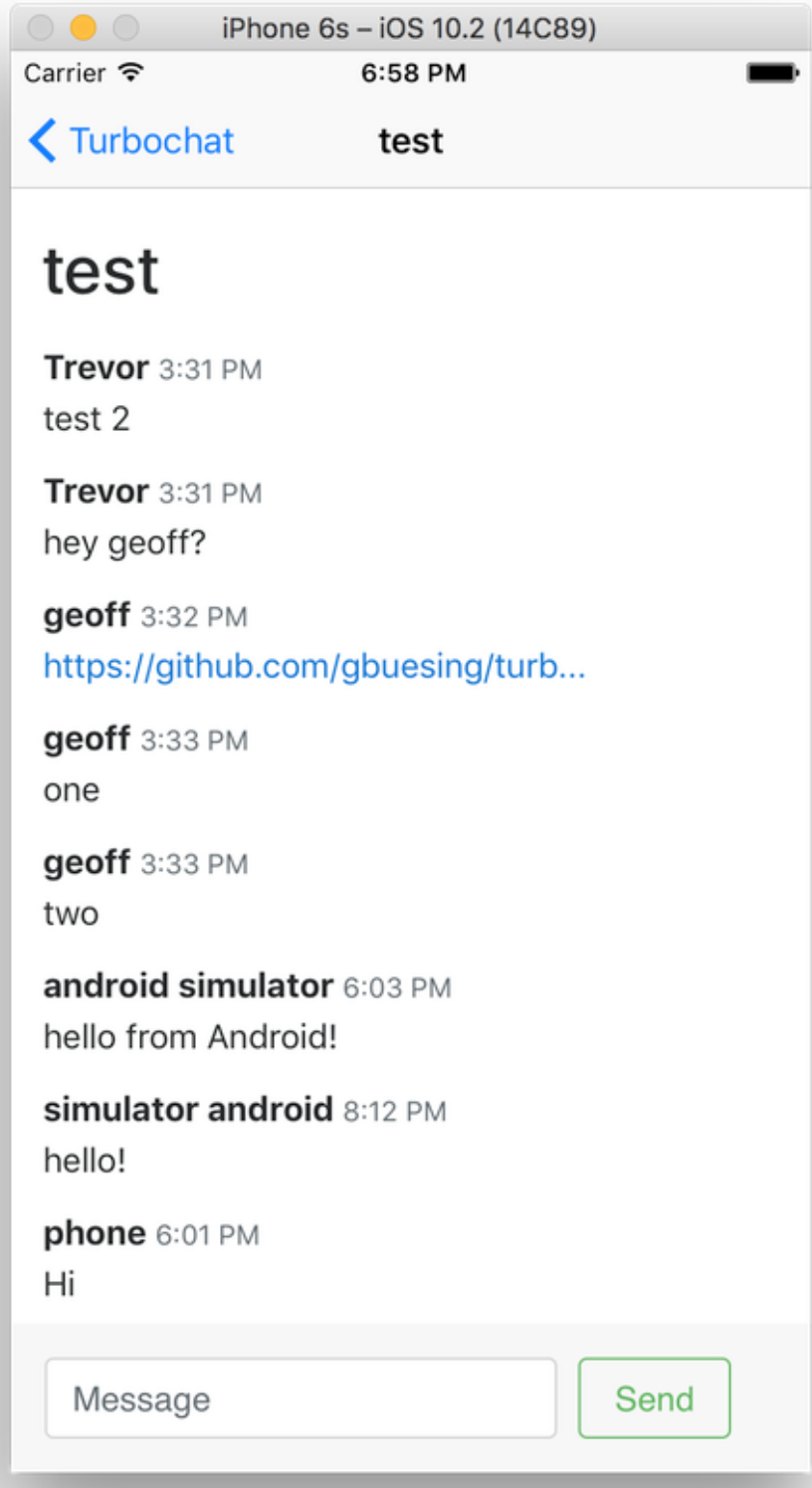
Android calls web app function

```
String js = "(function() { return new Date() })();";

ValueCallback callback = new ValueCallback<String>() {
    @Override
    public void onReceiveValue(String s) {
        Log.d("Date is:", s);
    }
};

webView.evaluateJavascript(js, callback);
```

Use native code and navigation UI progressively where needed

- Intercept links and render select views as native
- Use completely native views for pages that need native speed


iPhone 6s – iOS 10.2 (14C89)
Carrier
6:58 PM
Turbochat
Chat rooms
test
another
new room
android
new room'
Signed in as simulator | Sign out
New chat room
Room name


iPhone 6s – iOS 10.2 (14C89)
Carrier
6:58 PM
Turbochat
test
test
Trevor 3:31 PM
test 2
Trevor 3:31 PM
hey geoff?
geoff 3:32 PM
https://github.com/gbuesing/turb...
geoff 3:33 PM
one
geoff 3:33 PM
two
android simulator 6:03 PM
hello from Android!
simulator android 8:12 PM
hello!
phone 6:01 PM
Hi
Message
Send

Intercept a link and render native (iOS)

```
<a href="/rooms">Show me a native view</a>
```

```
// swift
func session(session: Session, didProposeVisitToURL
    URL: NSURL, withAction action: Action) {
    if URL.path == "/rooms" {
        // render native view
    } else {
        presentVisitableForSession(session, URL: URL,
            action: action)
    }
}
```

Intercept a link and render native (Android)

```
<a href="/rooms">Show me a native view</a>
```

```
// java
@Override
public void visitProposedToLocationWithAction(
        String location, String action) {
    if (location == 'https://myapp.com/rooms') {
        // render native view
    } else {
        Intent intent = new Intent(this,
                MainActivity.class);
        intent.putExtra(INTENT_URL, location);
        this.startActivity(intent);
    }
}
```

Conclusion

Pros

- Use mostly web tech you know (much easier to build adaptive views in HTML vs. native constraint layouts)
- Publish apps in the App Store and Play Store
- Deploy most bug fixes and new features for your web app without going through the store

Cons

- There's no magic, you'll need to learn Xcode and Android Studio
- ...and Swift and Java

Questions?

Geoff Buesing @gbuesing

Trevor Turk @trevorturk