So. Many. Decisions.

- Editor
  - Which one?
  - Which plugins?
  - Use built in terminal?
  - Editor config
- Module format
  - ES6 Modules, CommonJS...
- HTML generation
  - Minify?
  - Use plugin?
  - Inject prod only concerns?
  - Templating language?
- Transpiling
  - Native ES or diff language?
  - Use experimental features?
  - Which plugins?
  - Production vs dev config
- Bundler
  - Webpack, Browserify, Rollup...
- Linting
  - Which linter?
  - Enable which rules?
  - Warning or error?
  - Which plugins?
  - Use a preset?
- Testing
  - Framework?
  - Assertion Library?
  - Helpers?
  - Test file location?
  - File naming?
  - What environment?
  - Mocking?
  - Code Coverage
  - Continuous Integration
- Project structure
  - By file type or feature?
  - Centralize API?
  - Allow Inline JS?
  - Extract to POJOs?
- HTTP
  - Library
  - Mock schema format
  - Mock data generation
  - Mock server
- Production build
  - Minification
  - Sourcemaps
  - Bundle splitting
  - Cache busting
  - Error logging

Boilerplate	Build your own
Save time	Freedom
Proven	Perfect fit for your team
Full-featured	No unnecessary complexity
	Understand it
	Easier to change

Why a Starter Kit?

Codifies

- Decisions
- Best practices
- Lessons learned

Encourages consistency

Avoids forgetting important details

Increases quality

- Doing the “right” thing is the easy thing

Avoids repeating work

What Belongs in Your Starter Kit?

- Package Management
- Bundling
- Minification
- Sourcemaps
- Transpiling
- Dynamic HTML Generation
- Centralized HTTP
- Mock API framework
- Component libraries
- Development Webserver
- Linting
- Automated testing
- Continuous Integration
- Automated build
- Automated deployment
- Working example app