AMP First Boilerplate ⚡

This example sets up the boilerplate for an AMP First Site. You can see a live version here. The boilerplate includes the following features:

- AMP Extension helpers (`amp-state` , `amp-script` , ...)
- AMP Serviceworker integration
- App manifest
- Offline page

Deploy your own

Deploy the example using Vercel:

▲ Deploy

How to use

Execute `create-next-app` with npm or Yarn to bootstrap the example:

```
npx create-next-app --example amp-first amp-first-app
# or
yarn create next-app --example amp-first amp-first-app
# or
pnpm create next-app -- --example amp-first amp-first-app
```

Open http://localhost:3000 to view it in the browser. The page will reload if you make edits. You will also see AMP validation errors in the console.

Todo

Things you need to do after installing the boilerplate:

1. Update your app manifest in `public/manifest.json` with your app-specific data (`theme_color` , `background_color` , `name` , `short_name`).
2. Update the `THEME_COLOR` variable defined in `components/Layout.js` .
3. Update favicon and icons in `public/favicon.ico` and `public/static/images/icons-*.png` .
4. Set the default language in `pages/_document.js` .
5. Review the AMP Serviceworker implementation in `public/serviceworker.js` .

Tips & Tricks

- **Using AMP Components:** you can import AMP components using `next/head` . Checkout `components/amp/AmpCustomElement` for a simple way to import AMP components. Once the component is imported, you can use it like any other HTML element.

- **amp-bind & amp-state:** it's no problem to use `amp-bind` and `amp-state` with Next.js. There are two things to be aware of:

1. The `[...]` binding syntax, e.g. `[text]="myStateVariable"`, is not supported in JSX. Use `data-amp-bind-text="myStateVariable"` instead.

2. Initializing `amp-state` via JSON string is not supported in JSX:

   ```
   <amp-state id="myState">
     <script type="application/json">
       {
         "foo": "bar"
       }
     </script>
   </amp-state>
   ```

   Instead you need to use `dangerouslySetInnerHTML` to initialize the string. can use the `/components/amp/AmpState.js` component to see how it works. The same approach works for `amp-access` and `amp-consent` as well:

   ```
   <AmpState id="message" value={ message: 'Hello World' }/>
   ```

- **amp-list & amp-mustache:** mustache templates conflict with JSX and it's template literals need to be escaped. A simple approach is to escape them via back ticks: `` src={`{{imageUrl}}`} ``.

- **amp-script:** you can use amp-script to add custom JavaScript to your AMP pages. The boilerplate includes a helper `components/amp/AmpScript.js` to simplify using `amp-script`. The helper also supports embedding inline scripts. Good to know: Next.js uses AMP Optimizer under the hood, which automatically adds the needed script hashes for inline amp-scripts.

Deployment

For production builds, you need to run (the app will be build into the `.next` folder):

```
$ yarn build
```

To start the application in production mode, run:

```
$ yarn start
```

Deploy it to the cloud with Vercel (Documentation).