

# getInitialProps

**Note:** `getInitialProps` is a legacy API. We recommend using `getStaticProps` or `getServerSideProps` instead.

`getInitialProps` is an `async` function that can be added to the default exported React component for the page. It will run on both the server-side and again on the client-side during page transitions. The result of the function will be forwarded to the React component as `props`.

 `pages/index.tsx`




```

1  import { NextPageContext } from 'next';
2
3  Page.getInitialProps = async (ctx: NextPageContext) => {
4    const res = await fetch('https://api.github.com/repos/vercel/next.js');
5    const json = await res.json();
6    return { stars: json.stargazers_count };
7  };
8
9  export default function Page({ stars }: { stars: number }) {
10    return stars;
11  }

```

## Good to know:

- Data returned from `getInitialProps` is serialized when server rendering. Ensure the returned object from `getInitialProps` is a plain `Object`, and not using `Date`, `Map` or `Set`.
- For the initial page load, `getInitialProps` will run on the server only. `getInitialProps` will then also run on the client when navigating to a different route with the `next/link` component or by using `next/router`.
- If `getInitialProps` is used in a custom `_app.js`, and the page being navigated to is using `getServerSideProps`, then `getInitialProps` will also run on the server.

## Context Object

`getInitialProps` receives a single argument called `context`, which is an object with the following properties:

Name	Description
<code>pathname</code>	Current route, the path of the page in <code>/pages</code>
<code>query</code>	Query string of the URL, parsed as an object
<code>asPath</code>	<code>String</code> of the actual path (including the query) shown in the browser
<code>req</code>	<a href="#">HTTP request object</a> <sup>↗</sup> (server only)
<code>res</code>	<a href="#">HTTP response object</a> <sup>↗</sup> (server only)
<code>err</code>	Error object if any error is encountered during the rendering

## Caveats

- `getInitialProps` can only be used in `pages/` top level files, and not in nested components. To have nested data fetching at the component level, consider exploring the [App Router](#).
- Regardless of whether your route is static or dynamic, any data returned from `getInitialProps` as `props` will be able to be examined on the client-side in the initial HTML. This is to allow the page to be [hydrated](#) <sup>↗</sup> correctly. Make sure that you don't pass any sensitive information that shouldn't be available on the client in `props`.