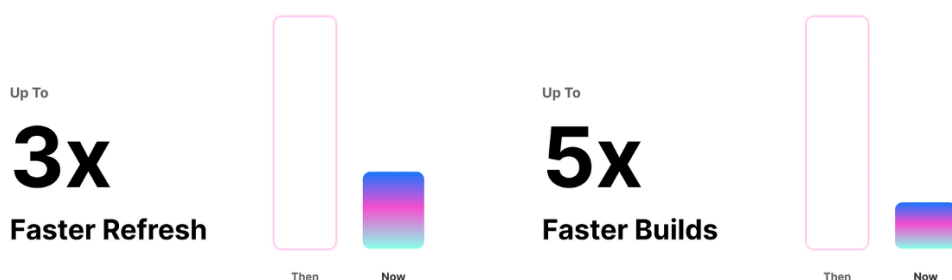


Next JS has announced at their Next.js conference that **Next JS 12** is going to be one of their biggest releases ever. So in this article, let's quickly look into what are the new amazing features Next.js 12 has to provide us with.

There are roughly 6 new features that Next.js 12 has brought out, and let's look into all of them.

1. Rust Compiler

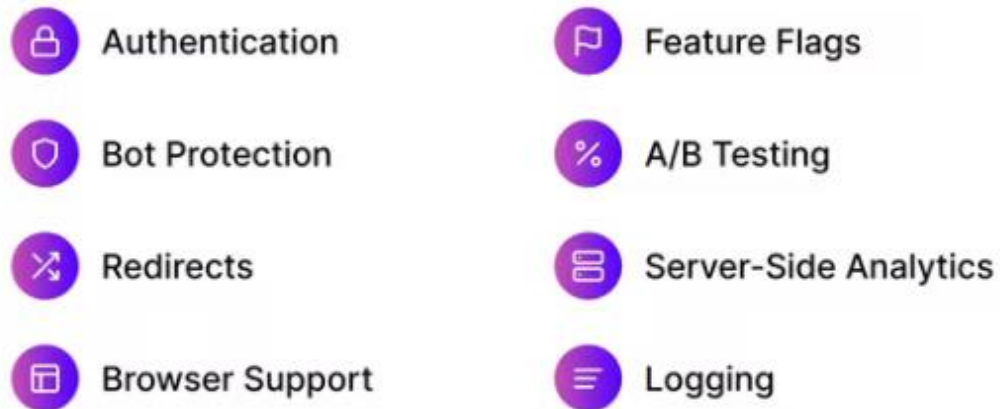
You really don't need to know much about Rust, but what you should know is that Next.js 12 is making use of Rust Compiler to provide **5x** faster builds and faster refresh, which was the previously **3x** the speed. This means that the performance has increased almost twice.



2. Middlewares

Middleware enables you to use code over configuration. This gives you full flexibility in Next.js because you can run code before a request is completed. Based on the user's incoming request, you can modify the response by rewriting, redirecting, adding headers, or even streaming HTML.

Middleware



3. React 18

We already might have heard about React 18 and its amazing features such as *suspense*, *automatic batching of updates*, *APIs like `startTransition`*, and *a new streaming API for server rendering with support for `React.lazy`*. **Next.js 12** will be adding these features as well.



4. Smaller images using AVIF

The built-in Image Optimization API now supports AVIF images, enabling 20% smaller images compared to WebP.

AVIF images can take longer to optimize compared to WebP, so we're making this feature opt-in using the new `images.formats` property in `next.config.js`:

```
module.exports = {
  images: {
    formats: ['image/avif', 'image/webp']
  }
}
```

5. Bot-Aware ISR Fallback

Currently, [Incremental Static Regeneration](#) with `fallback: true` renders a fallback state before rendering the page contents on the first request to a page that was not generated yet. To block the page from loading (server-rendering), you would need to use `fallback: 'blocking'`.

In Next.js 12, [web crawlers \(e.g. search bots\)](#) will automatically server-render ISR pages using `fallback: true`, while still serving the previous behavior of the fallback state to non-crawler User-Agents. This prevents crawlers from indexing loading states.

6. URL Imports

URL imports allow you to use *any* package directly through a URL. This enables Next.js to process remote HTTP(S) resources exactly like local dependencies.

If a URL import is detected, Next.js will generate a `next.lock` file to track remote resources. URL imports are cached locally to ensure you can still work offline. Next.js supports both client and server URL imports.

To opt-in, add the allowed URL prefixes inside `next.config.js`:

```
module.exports = {
  experimental: {
    urlImports: ['https://cdn.skypack.dev']
  }
}
```

Then, you can import modules directly from URLs:

```
import confetti from 'https://cdn.skypack.dev/canvas-confetti'
```

Here are the 6 new features of Next.js, but always be up-to-date with the Next JS documentation for any other latest updates as well.

Happy learning 😊