

Multi-Zone Next.js Cheat Sheet

1. Zones & Host

- **Zones:** Separate Next.js apps, e.g., Auth, Events, Dashboard.
- **Host:** Main app that forwards requests using rewrites.
- **User sees** only unified domain (`myapp.com/events`), not ports or subdomains.
- **Example rewrites** (`next.config.js`):

```
rewrites: async () => [  
  { source: '/auth/:path*', destination: 'http://localhost:3001/:path*' },  
  { source: '/events/:path*', destination: 'http://localhost:  
3002/:path*' },  
  { source: '/dashboard/:path*', destination: 'http://localhost:  
3003/:path*' },  
];
```

2. Shared API Client

```
import axios from 'axios';  
  
const api = axios.create({  
  baseURL: '/', // always through host  
  withCredentials: true, // send cookies automatically  
});  
  
api.interceptors.response.use(  
  res => res,  
  err => {  
    if (err.response?.status === 401) {  
      window.location.href = '/auth/login';  
    }  
    return Promise.reject(err);  
  }  
);  
  
export default api;
```

- **Base URL** `/`: ensures request goes through host. - **withCredentials**: sends JWT cookie automatically. - **Interceptors**: handle 401 unauthorized globally.

3. Cross-Zone Navigation

- Host contains main navigation:

```
<Link href='/events'>Events</Link>
<Link href='/dashboard'>Dashboard</Link>
```

- **Prefetch for smooth UX:** `<Link href='/events' prefetch>`
- **State passing:** query params, sessionStorage, or localStorage.
- **Shared layout:** host provides header/footer, zones provide content.

4. Auth Flow

Login API (Auth Zone)

```
res.cookies.set('auth_token', token, {
  httpOnly: true,
  secure: true, // production
  sameSite: 'lax',
  path: '/', // accessible by all zones
  maxAge: 3600,
});
```

Middleware (Protect Zones)

```
export function middleware(req: NextRequest) {
  const token = req.cookies.get('auth_token')?.value;
  if (!token) return NextResponse.redirect('/auth/login');
  try { jwt.verify(token, secret); return NextResponse.next(); }
  catch { return NextResponse.redirect('/auth/login'); }
}

export const config = {
  matcher: ['/events/:path*', '/dashboard/:path*'],
};
```

- **Matcher:** paths to protect (middleware only runs here).
- **JWT cookie Path=/:** accessible by all zones.
- **Login form posts to** `/auth/api/login` → cookie set automatically.

5. Deployment

Vercel

- Separate projects → host rewrites to subdomains.
- Monorepo → folders + rewrites.
- Cookies must be domain-wide.

Nginx

```
location /auth/ { proxy_pass http://127.0.0.1:3001; }  
location /events/ { proxy_pass http://127.0.0.1:3002; }  
location /dashboard/ { proxy_pass http://127.0.0.1:3003; }
```

- Browser only sees host domain.

AWS / ECS

- Containers per zone.
- Host routes via CloudFront / ALB.
- Cookies + API calls preserved.

6. SSR (Server-Side Rendering)

```
export async function getServerSideProps(context) {  
  const token = context.req.cookies.auth_token;  
  if (!token) return { redirect: { destination: '/auth/login', permanent:  
false } };  
  
  const res = await fetch('https://api.backend.com/events', {  
    headers: { cookie: context.req.headers.cookie || '' }  
  });  
  const events = await res.json();  
  
  return { props: { events } };  
}
```

- Server checks cookie before rendering. - Data is pre-fetched → faster first paint. - Works seamlessly across zones.

7. Key Tips

1. **Always use host paths** for API calls (`/api/...`).
2. **Use one unified domain** if possible (simplifies cookies).
3. **Middleware + matcher** protects only private paths.
4. **Shared API client** centralizes auth handling.
5. **Prefetch and shared layout** make zones feel unified.
6. **SSR + cookies** ensure auth works on first load.

This cheat sheet covers **everything you need to build a production-ready multi-zone Next.js app** with auth, API calls, cross-navigation, SSR, and deployment.