```
/* CONFIGURATION STARTS HERE */

/* Step 1: enter your domain name like fruitionsite.com */
const MY_DOMAIN = 'thumos.life';

/*
* Step 2: enter your URL slug to page ID mapping
* The key on the left is the slug (without the slash)
* The value on the right is the Notion page ID
*/
const SLUG_TO_PAGE = {
  '': 'fd78e23319ea4efb82e8313c7e93b3a2',
};

/* Step 3: enter your page title and description for SEO purposes */
const PAGE_TITLE = 'Thumos.Life';
const PAGE_DESCRIPTION = 'Home on the interweb for me to ramble about values
and such';

/* Step 4: enter a Google Font name, you can choose from https://fonts.google.com */
const GOOGLE_FONT = '';

/* Step 5: enter any custom scripts you'd like */
const CUSTOM_SCRIPT = ``;

/* CONFIGURATION ENDS HERE */

const PAGE_TO_SLUG = {};
const slugs = [];
const pages = [];
Object.keys(SLUG_TO_PAGE).forEach(slug => {
  const page = SLUG_TO_PAGE[slug];
  slugs.push(slug);
  pages.push(page);
  PAGE_TO_SLUG[page] = slug;
});

addEventListener('fetch', event => {
  event.respondWith(fetchAndApply(event.request));
});

function generateSitemap() {
  let sitemap = '<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">';
  slugs.forEach(
    (slug) =>
      (sitemap +=
        '<url><loc>https://' + MY_DOMAIN + '/' + slug + '</loc></url>')
```

```
  );
  sitemap += '</urlset>';
  return sitemap;
}

const corsHeaders = {
  'Access-Control-Allow-Origin': '*',
  'Access-Control-Allow-Methods': 'GET, HEAD, POST, PUT, OPTIONS',
  'Access-Control-Allow-Headers': 'Content-Type',
};

function handleOptions(request) {
  if (request.headers.get('Origin') !== null &&
    request.headers.get('Access-Control-Request-Method') !== null &&
    request.headers.get('Access-Control-Request-Headers') !== null) {
    // Handle CORS pre-flight request.
    return new Response(null, {
      headers: corsHeaders
    });
  } else {
    // Handle standard OPTIONS request.
    return new Response(null, {
      headers: {
        'Allow': 'GET, HEAD, POST, PUT, OPTIONS',
      }
    });
  }
}

async function fetchAndApply(request) {
  if (request.method === 'OPTIONS') {
    return handleOptions(request);
  }
  let url = new URL(request.url);
  if (url.pathname === '/robots.txt') {
    return new Response('Sitemap: https://' + MY_DOMAIN + '/sitemap.xml');
  }
  if (url.pathname === '/sitemap.xml') {
    let response = new Response(generateSitemap());
    response.headers.set('content-type', 'application/xml');
    return response;
  }
  const notionUrl = 'https://www.notion.so' + url.pathname;
  let response;
  if (url.pathname.startsWith('/app') && url.pathname.endsWith('js')) {
    response = await fetch(notionUrl);
    let body = await response.text();
```

```
    response = new Response(body.replace(/www.notion.so/g, MY_DOMAIN).replace(/
notion.so/g, MY_DOMAIN), response);
    response.headers.set('Content-Type', 'application/x-javascript');
  } else if ((url.pathname.startsWith('/api'))) {
    // Forward API
    response = await fetch(notionUrl, {
      body: request.body,
      headers: {
        'content-type': 'application/json;charset=UTF-8',
        'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36'
      },
      method: 'POST',
    });
    response = new Response(response.body, response);
    response.headers.set('Access-Control-Allow-Origin', '*');
  } else if (slugs.indexOf(url.pathname.slice(1)) > -1) {
    const pageId = SLUG_TO_PAGE[url.pathname.slice(1)];
    return Response.redirect('https://' + MY_DOMAIN + '/' + pageId, 301);
  } else {
    response = await fetch(notionUrl, {
      body: request.body,
      headers: request.headers,
      method: request.method,
    });
    response = new Response(response.body, response);
    response.headers.delete('Content-Security-Policy');
    response.headers.delete('X-Content-Security-Policy');
  }

  return appendJavascript(response, SLUG_TO_PAGE);
}

class MetaRewriter {
  element(element) {
    if (PAGE_TITLE !== '') {
      if (element.getAttribute('property') === 'og:title'
        || element.getAttribute('name') === 'twitter:title') {
        element.setAttribute('content', PAGE_TITLE);
      }
      if (element.tagName === 'title') {
        element.setInnerContent(PAGE_TITLE);
      }
    }
    if (PAGE_DESCRIPTION !== '') {
      if (element.getAttribute('name') === 'description'
        || element.getAttribute('property') === 'og:description'
```

```
        ll element.getAttribute('name') === 'twitter:description') {
        element.setAttribute('content', PAGE_DESCRIPTION);
      }
    }
    if (element.getAttribute('property') === 'og:url'
      ll element.getAttribute('name') === 'twitter:url') {
      element.setAttribute('content', MY_DOMAIN);
    }
    if (element.getAttribute('name') === 'apple-itunes-app') {
      element.remove();
    }
  }
}

class HeadRewriter {
  element(element) {
    if (GOOGLE_FONT !== '') {
      element.append(`<link href="https://fonts.googleapis.com/css?family=$
{GOOGLE_FONT.replace(' ', '+')}:Regular,Bold,Italic&display=swap" rel="stylesheet">
      <style>* { font-family: "${GOOGLE_FONT}" !important; }</style>`, {
       html: true
      });
    }
    element.append(`<style>
    div.notion-topbar > div > div:nth-child(3) { display: none !important; }
    div.notion-topbar > div > div:nth-child(4) { display: none !important; }
    div.notion-topbar > div > div:nth-child(5) { display: none !important; }
    div.notion-topbar > div > div:nth-child(6) { display: none !important; }
    div.notion-topbar-mobile > div:nth-child(3) { display: none !important; }
    div.notion-topbar-mobile > div:nth-child(4) { display: none !important; }
    </style>`, {
      html: true
    })
  }
}

class BodyRewriter {
  constructor(SLUG_TO_PAGE) {
    this.SLUG_TO_PAGE = SLUG_TO_PAGE;
  }
  element(element) {
    element.append(`<div style="display:none">Powered by <a href="http://
fruitionsite.com">Fruition</a></div>
    <script>
    const SLUG_TO_PAGE = ${JSON.stringify(this.SLUG_TO_PAGE)};
    const PAGE_TO_SLUG = {};
    const slugs = [];
```

```
const pages = [];
let redirected = false;
Object.keys(SLUG_TO_PAGE).forEach(slug => {
  const page = SLUG_TO_PAGE[slug];
  slugs.push(slug);
  pages.push(page);
  PAGE_TO_SLUG[page] = slug;
});
function getPage() {
  return location.pathname.slice(-32);
}
function getSlug() {
  return location.pathname.slice(1);
}
function updateSlug() {
  const slug = PAGE_TO_SLUG[getPage()];
  if (slug != null) {
    history.replaceState(history.state, '', '/' + slug);
  }
}
const observer = new MutationObserver(function() {
  if (redirected) return;
  const nav = document.querySelector('.notion-topbar');
  const mobileNav = document.querySelector('.notion-topbar-mobile');
  if (nav && nav.firstChild && nav.firstChild.firstChild
    || mobileNav && mobileNav.firstChild) {
    redirected = true;
    updateSlug();
    const onpopstate = window.onpopstate;
    window.onpopstate = function() {
      if (slugs.includes(getSlug())) {
        const page = SLUG_TO_PAGE[getSlug()];
        if (page) {
          history.replaceState(history.state, 'bypass', '/' + page);
        }
      }
      onpopstate.apply(this, [].slice.call(arguments));
      updateSlug();
    };
  }
});
observer.observe(document.querySelector('#notion-app'), {
  childList: true,
  subtree: true,
});
const replaceState = window.history.replaceState;
window.history.replaceState = function(state) {
```

```
      if (arguments[1] !== 'bypass' && slugs.includes(getSlug())) return;
      return replaceState.apply(window.history, arguments);
    };
    const pushState = window.history.pushState;
    window.history.pushState = function(state) {
      const dest = new URL(location.protocol + location.host + arguments[2]);
      const id = dest.pathname.slice(-32);
      if (pages.includes(id)) {
        arguments[2] = '/' + PAGE_TO_SLUG[id];
      }
      return pushState.apply(window.history, arguments);
    };
    const open = window.XMLHttpRequest.prototype.open;
    window.XMLHttpRequest.prototype.open = function() {
      arguments[1] = arguments[1].replace('${MY_DOMAIN}', 'www.notion.so');
      return open.apply(this, [].slice.call(arguments));
    };
  </script>${CUSTOM_SCRIPT}`, {
      html: true
    });
  }
}

async function appendJavascript(res, SLUG_TO_PAGE) {
  return new HTMLRewriter()
    .on('title', new MetaRewriter())
    .on('meta', new MetaRewriter())
    .on('head', new HeadRewriter())
    .on('body', new BodyRewriter(SLUG_TO_PAGE))
    .transform(res);
}
```