

# Temporary Download URL

```
import crypto from 'crypto';

// Generate a signed URL valid for 5 minutes
function generateSignedUrl(fileName: string, secret: string) {
  const expiresIn = 5 * 60 * 1000; // 5 minutes
  const expiresAt = Date.now() + expiresIn;
  const signature = crypto.createHmac('sha256', secret)
    .update(`${fileName}:${expiresAt}`)
    .digest('hex');

  return `/download/${fileName}?expiresAt=${expiresAt}&signature=${signature}`;
}

// Verify the signed URL on request
app.get('/download/:fileName', (req: Request, res: Response) => {
  const { fileName } = req.params;
  const { expiresAt, signature } = req.query;

  if (!expiresAt || !signature) {
    return res.status(401).json({ message: 'Invalid download link' });
  }

  const secret = 'your_secret_key';
  const expectedSignature = crypto.createHmac('sha256', secret)
    .update(`${fileName}:${expiresAt}`)
    .digest('hex');

  if (signature !== expectedSignature || Date.now() > Number(expiresAt)) {
    return res.status(403).json({ message: 'Link expired or invalid' });
  }

  // Proceed with file download if the signature is valid and not expired
  const filePath = path.join(__dirname, 'uploads', fileName);
  res.download(filePath);
});
```

Revision #1

Created 6 November 2024 11:04:39 by Makito

Updated 6 November 2024 11:05:52 by Makito