

# Bruno API Collection: Resume Optimization

This directory contains a comprehensive API collection for the Resume Optimization Service built with [Bruno](#), a lightweight API client for testing and documentation.

## Table of Contents

- [Overview](#)
  - [Available Endpoints](#)
  - [Getting Started](#)
  - [Configuration](#)
  - [Request Examples](#)
  - [Response Format](#)
  - [Error Handling](#)
  - [Directory Structure](#)
  - [Tips & Best Practices](#)
  - [Troubleshooting](#)
  - [Additional Resources](#)
- 

## Overview

These request definitions enable seamless testing and integration with the resume optimization backend API. Each request is pre-configured with proper endpoints, authentication, and example payloads.

---

## Available Endpoints

### 1. Optimize Resume (`OptimizeResume.bru`)

- **Method:** POST
- **Endpoint:** /api/upload
- **Purpose:** Upload a resume and job description for AI-powered optimization
- **Parameters:**
  - `resume`: Resume file (multipart)
  - `job`: Job description file (multipart)
  - `optimize`: JSON configuration object with `resume` and `jobDescription` fields

### 2. List Files (`ListFiles.bru`)

- **Method:** GET
- **Endpoint:** /api/files
- **Purpose:** Retrieve a list of all uploaded and generated files with metadata

### 3. Download File (`DownloadFile.bru`)

- **Method:** GET
- **Endpoint:** /api/files/{filename}

- **Purpose:** Download a specific file by name

## 4. Delete File (**DeleteFile.bru**)

- **Method:** DELETE
- **Endpoint:** /api/files/{filename}
- **Purpose:** Remove a file from the server

## 5. Convert Markdown to PDF (**Convert Markdown to PDF.bru**)

- **Method:** POST
  - **Endpoint:** /api/markdownFile2PDF
  - **Purpose:** Convert a Markdown document to PDF format
- 

# Getting Started

## Prerequisites

- [Bruno](#) installed on your system
- Backend API running at `http://localhost:8080`

## Running the Requests

1. Open Bruno and load this collection
2. Select the desired request from the sidebar
3. Configure parameters as needed (file uploads, form data, etc.)
4. Click "Send" to execute the request
5. View the response in the result panel

# Configuration

All requests are pre-configured with:

- **Base URL:** `http://localhost:8080`
- **CORS:** Properly configured for local development
- **Authentication:** None (unsecured endpoints for local testing)

To change the base URL for different environments (development, staging, production), update the hostname in each request's URL field.

---

# Request Examples

## Example: Optimize a Resume

1. Open `OptimizeResume.bru`
2. Attach your resume file to the `resume` field
3. Attach a job description file to the `job` field
4. Optionally configure the `optimize` JSON with custom settings

5. Send the request
6. Response will be "generating" with HTTP 200 OK

## Example: List All Files

1. Open ListFiles.bru
  2. Click Send
  3. Receive a JSON array of all available files with download URLs and timestamps
- 

## Response Format

All successful responses follow this format:

```
{  
  "message": "Response message describing the result",  
  "status": 200  
}
```

For file operations, responses may include:

- File metadata (name, URL, creation date)
  - File list with multiple entries
  - Generated file paths
- 

## Error Handling

Common HTTP status codes:

- **200 OK:** Request successful
  - **400 Bad Request:** Invalid parameters or missing required fields
  - **404 Not Found:** File not found
  - **500 Internal Server Error:** Server-side processing error
- 

## Directory Structure

```
bruno/Resume/  
  README.md  
  OptimizeResume.bru          # This file  
  ListFiles.bru               # Resume optimization request  
  DownloadFile.bru            # List all files request  
  DeleteFile.bru              # Download file request  
  Convert Markdown to PDF.bru # Delete file request  
                            # Markdown to PDF conversion request
```

---

## Tips & Best Practices

- **File Naming:** Use descriptive filenames with appropriate extensions (.pdf, .txt, .md)
- **Batch Operations:** Use ListFiles to verify uploads before further processing
- **Testing Flow:** Optimize List Download to verify the complete workflow

- **Performance:** Large files may take time to process; check HTTP response codes
- 

## Troubleshooting

- **Connection Refused:** Ensure the backend API is running on port 8080
  - **File Not Found:** Verify the filename exactly matches the listing from ListFiles
  - **CORS Errors:** Check that your frontend is running on an allowed origin (3000, 5173, 80)
- 

## Additional Resources

- [Bruno Documentation](#)
  - [REST API Best Practices](#)
  - [Project Repository](#)
- 

**Version:** 1.0 **Last Updated:** January 17, 2026 **Maintained By:** Development Team