

Setup & Authentication

Installation

```
$ npm install -g wrangler
$ yarn global add wrangler
```

Authentication

```
$ wrangler login
```

Login via browser

```
$ wrangler login --api-token TOKEN
```

Use API token directly

```
$ wrangler whoami
```

Check auth status

Development

Project Init

```
$ wrangler init my-worker
$ wrangler init --type typescript
```

Local Development

```
$ wrangler dev
```

Start dev server

```
$ wrangler dev --port 8787
$ wrangler dev --local
$ wrangler dev --remote
```

Use --local for speed, --remote for edge simulation

Deployment

Basic Deploy

```
$ wrangler deploy
```

Deploy to production

```
$ wrangler deploy --env staging
```

Deploy to environment

```
$ wrangler deploy --dry-run
```

Preview deployment

Management

```
$ wrangler deployments list
$ wrangler rollback
$ wrangler delete my-worker
```

KV Storage

Namespace Management

```
$ wrangler kv:namespace create "MY_KV"
$ wrangler kv:namespace list
$ wrangler kv:namespace delete --binding MY_KV
```

Data Operations

```
$ wrangler kv:key list --binding MY_KV
$ wrangler kv:key get "key" --binding MY_KV
$ wrangler kv:key put "key" "value" --binding MY_KV
$ wrangler kv:key delete "key" --binding MY_KV
```

Bulk Operations

```
$ wrangler kv:bulk put data.json --binding MY_KV
$ wrangler kv:bulk delete keys.json --binding MY_KV
```

R2 Storage

Bucket Management

```
$ wrangler r2:bucket create my-bucket
$ wrangler r2:bucket list
$ wrangler r2:bucket delete my-bucket
```

Object Operations

```
$ wrangler r2:object put bucket/file.txt --file ./local.txt
$ wrangler r2:object get bucket/file.txt --file ./download.txt
$ wrangler r2:object list my-bucket
$ wrangler r2:object delete bucket/file.txt
```

D1 Database

Database Management

```
$ wrangler d1 create my-database
$ wrangler d1 list
$ wrangler d1 delete my-database
```

SQL Operations

```
$ wrangler d1 execute my-db --command "SELECT * FROM users"
$ wrangler d1 execute my-db --file schema.sql
```

Migrations

```
$ wrangler d1 migrations create my-db "add-users-table"
$ wrangler d1 migrations apply my-db
$ wrangler d1 migrations list my-db
```

Pages

Project Management

```
$ wrangler pages project create my-site
$ wrangler pages project list
$ wrangler pages project delete my-site
```

Deployment

```
$ wrangler pages deploy ./dist
$ wrangler pages deploy ./dist --project-name my-site
```

Functions

```
$ wrangler pages functions build
$ wrangler pages dev ./dist
```

Workers Management

Worker Operations

```
$ wrangler deploy
```

Deploy worker to production

```
$ wrangler deploy --name my-custom-worker
$ wrangler deploy --env staging
```

Worker Info

```
$ wrangler list
```

List all workers

```
$ wrangler delete my-worker
$ wrangler subdomain get
$ wrangler subdomain set my-subdomain
```

Configuration

```
# wrangler.toml name = "my-worker" main = "src/index.ts" compatibility_date = "2024-01-01" [build] command = "npm run build"
```

Monitoring & Logs

Real-time Logs

```
$ wrangler tail
```

View live logs

```
$ wrangler tail --status error
$ wrangler tail --method POST
$ wrangler tail --search "keyword"
```

Analytics

```
$ wrangler analytics
$ wrangler pages deployment tail
```

Use filters: --status, --method, --search

Secrets & Env

Secrets Management

```
$ wrangler secret put SECRET_KEY
$ wrangler secret list
$ wrangler secret delete SECRET_KEY
```

Environment Variables

```
$ wrangler secret put KEY --env staging
$ wrangler pages secret put KEY --project-name my-site
```

```
# wrangler.toml [vars] ENVIRONMENT = "production" API_URL = "https://api.example.com" [[kv_namespaces]] binding = "MY_KV" id = "namespace-id"
```

Routes & Domains

Route Management

```
$ wrangler route add "api.example.com/*" my-worker
$ wrangler route list
$ wrangler route delete route-id
```

Custom Domains

```
$ wrangler custom-domains list
$ wrangler custom-domains create example.com
```

Routes take precedence over custom domains

Utilities

Validation & Info

```
$ wrangler validate
```

Validate wrangler.toml

```
$ wrangler --version
$ wrangler --help
```

Debugging

```
$ WRANGLER_LOG=debug wrangler dev
$ wrangler dev --log-level debug
```

Cache & Cleanup

```
$ rm -rf node_modules .wrangler
```

Clear build cache

Set WRANGLER\_LOG=debug for detailed logging