

# Django Project Setup with Hot Reload - Complete Guide

## 1. Install Required Packages

First, install all the packages you need based on your settings.py:

```
bash

# Core Django packages
pip install django==5.2.3

# Environment variables
pip install python-dotenv

# REST Framework and JWT
pip install djangorestframework
pip install djangorestframework-simplejwt

# Channels for WebSockets
pip install channels
pip install channels-redis

# Email functionality
pip install yagmail

# Development tools with hot reload
pip install django-browser-reload
pip install django-debug-toolbar
pip install django-extensions
```

## 2. Updated settings.py Configuration

Add these to your existing settings.py:

python

*# Add to INSTALLED\_APPS*

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'backend',  
    'channels',  
    'rest_framework',  
    'rest_framework_simplejwt',  
    # Development tools (only add these)  
    'django_browser_reload',  
    'debug_toolbar',  
    'django_extensions',  
]
```

*# Update MIDDLEWARE*

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
    # Add these development middlewares  
    'django_browser_reload.middleware.BrowserReloadMiddleware',  
    'debug_toolbar.middleware.DebugToolbarMiddleware',  
]
```

*# Add at the bottom of your settings.py file*

*# Debug Toolbar Settings*

```
if DEBUG:  
    import socket  
    hostname, _, ips = socket.gethostbyname_ex(socket.gethostname())  
    INTERNAL_IPS = [ip[: ip.rfind(".")] + ".1" for ip in ips] + ["127.0.0.1", "10.0.2.2"]
```

### 3. Updated Main URLs Configuration

Replace your main `urls.py` with this:

```
python

from django.contrib import admin
from django.urls import path, include
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('backend.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

# Development URLs
if settings.DEBUG:
    urlpatterns = [
        path('__debug__/', include('debug_toolbar.urls')),
        path('__reload__/', include('django_browser_reload.urls')),
    ] + urlpatterns
```

## 4. Backend App URLs (keep as is)

Your `backend/urls.py` should remain:

```
python

from django.urls import path
from . import views

urlpatterns = [
    path('', views.homepage, name='homepage'),
]
```

## 5. Running the Development Server

For Windows PowerShell, use one of these methods:

### Method 1: Single Command (Recommended)

```
bash

python manage.py runserver
```

The `django-browser-reload` package works automatically with the regular runserver command.

## Method 2: Two Separate Terminals

If you want more control:

### Terminal 1:

```
bash

python manage.py runserver
```

### Terminal 2 (if needed):

```
bash

# This is optional - browser reload works automatically
python manage.py shell
```

## 6. How Hot Reload Works

With `django-browser-reload`:

- **Template changes:** Browser automatically refreshes
- **Python code changes:** Django's built-in auto-reloader restarts the server
- **Static file changes:** Browser automatically refreshes
- **CSS/JS changes:** Browser automatically refreshes

## 7. Verification Steps

### 1. Start the server:

```
bash

python manage.py runserver
```

### 2. Check the console output - you should see:

```
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
```

### 3. Visit your site at `http://127.0.0.1:8000/`

### 4. Test hot reload:

- Edit a template file
- Save the file
- Browser should automatically refresh

5. **Debug toolbar** should appear on your pages when `DEBUG = True`

## 8. Troubleshooting

**If you still get livereload errors:**

1. **Remove any old livereload packages:**

```
bash

pip uninstall django-livereload-server
pip uninstall django-livereload
```

2. **Clean your middleware** - remove any references to:

- `livereload.middleware.LiveReloadScript`
- `livereload.middleware.LiveReloadMiddleware`

3. **Use only the django-browser-reload package** as shown above

**If debug toolbar doesn't appear:**

- Make sure `DEBUG = True`
- Check that your IP is in `INTERNAL_IPS`
- Ensure templates have `<body>` tags (debug toolbar injects itself there)

## 9. Production Notes

All these development tools should be disabled in production:

- Set `DEBUG = False`
- Remove development apps from `INSTALLED_APPS`
- Remove development middleware
- Use environment variables to control this

## 10. Alternative: Using django-extensions

If you installed `django-extensions`, you can also use:

```
bash
```

```
python manage.py runserver_plus --reloader-type stat
```

This provides additional development features and better error pages.