

AI Marketer V2 Deployment Guide

Background

CORS issues may arise when both the frontend and backend are running locally, often due to misconfigurations, preventing proper communication between them. To bypass this issue, both the frontend and backend could be deployed to their respective production servers. However, deploying both may not be ideal for ongoing development. Therefore, it may be useful to run the part that's currently under development (either the frontend or backend) locally, while deploying the other to a server. This setup allows for continued development without CORS issues, while also enabling immediate visibility of code changes by running the locally modified part.

Frontend Deployment (Vercel)

The frontend application is hosted on Vercel at:

```
https://ai-marketer-v2-frontend.vercel.app/
```

Note: The Vercel URL may change depending on the actual deployment.

If this URL changes, make sure to update the corresponding values in your configuration.

Configuring Backend Connection on Local Frontend

To configure where your local frontend sends API requests:

1. Create a `.env.local` file in the frontend root directory
2. Add the following variable:

```
NEXT_PUBLIC_API_URL=https://backend-d6fz.onrender.com
```

- For Render backend: `https://backend-d6fz.onrender.com`
- For local backend: `https://localhost:8000`

Configuring Backend Connection on Vercel Frontend

To change the API endpoint for the Vercel-hosted frontend:

1. Go to the Vercel dashboard for your project
2. Navigate to Settings > Environment Variables
3. Add or update the `NEXT_PUBLIC_API_URL` variable
4. Set it to either:
 - `https://backend-d6fz.onrender.com` for Render backend
 - `https://localhost:8000` for local backend

Backend Deployment (Render)

The backend application is hosted on Render at:

```
https://backend-d6fz.onrender.com
```

Note: The Render URL may change depending on the actual deployment.

If this URL changes, make sure to update the corresponding values in your configuration.

Configuring CORS for Local Backend

To allow connections from both local and Vercel frontends to your local backend:

1. Add the following to your `.env` file:

```
CORS_ALLOWED_ORIGINS=https://localhost:3000,https://ai-marketer-v2-frontend.vercel.app
ALLOWED_HOSTS=localhost,127.0.0.1,0.0.0.0,ai-marketer-v2-frontend.vercel.app
```

Configuring CORS for Render Backend

To allow connections to your Render backend:

1. Go to the Render dashboard for your backend service
2. Navigate to the Environment section
3. Add or update the following variables:

```
ALLOWED_HOSTS=localhost,127.0.0.1,0.0.0.0,backend-d6fz.onrender.com
CORS_ALLOWED_ORIGINS=https://localhost:3000,https://ai-marketer-v2-frontend.vercel.app
```

This configuration ensures that your Render backend can accept connections from both local development environments and the production Vercel frontend.

Development Setup

Frontend Development

When working on the frontend, follow these steps:

1. **Configure the local frontend to connect to the remote backend**

To ensure your local frontend connects to the remote backend on Render, follow these steps:

- Create or modify the `.env.local` file in your frontend root directory.
- Add or update the `NEXT_PUBLIC_API_URL` variable to point to the remote backend:

```
NEXT_PUBLIC_API_URL=https://backend-d6fz.onrender.com
```

- **Note:** The backend URL may change, so be sure to update the `NEXT_PUBLIC_API_URL` value according to the actual backend URL.

2. Run the local frontend

Start the local frontend by running it at `https://localhost:3000`. This allows you to develop and test the frontend locally.

3. Test the frontend locally

After making changes to the frontend, you can test it locally while it communicates with the remote backend (on Render). This setup allows you to see real-time changes without deploying the frontend to production.

- **Test Logs:** To verify that the frontend is correctly communicating with the backend, check the Render logs for any incoming requests or responses. This can help ensure that the connection is properly established.

Backend Development

When working on the backend, follow these steps:

1. Configure the local backend to connect to the remote frontend

To ensure your local backend connects to the remote frontend on Vercel, follow these steps:

- Create or modify the `.env` file in your backend root directory.
- Add or update the `CORS_ALLOWED_ORIGINS` and `ALLOWED_HOSTS` variables to allow the frontend to connect:

```
CORS_ALLOWED_ORIGINS=https://localhost:3000,https://ai-marketer-v2-frontend.vercel.app
ALLOWED_HOSTS=localhost,127.0.0.1,0.0.0.0,ai-marketer-v2-frontend.vercel.app
```

- **Note:** Make sure to adjust these variables based on the actual URLs for your frontend.

2. Configure Vercel Frontend to Connect to the Local Backend

To ensure the remote frontend hosted on Vercel connects to your local backend, follow these steps:

- Go to your Vercel dashboard.
- Navigate to Settings > Environment Variables.
- Add or update the `NEXT_PUBLIC_API_URL` variable to point to your local backend:

```
NEXT_PUBLIC_API_URL=https://localhost:8000
```

3. Run the local backend

Start the local backend by running it at `https://localhost:8000`. This allows you to develop and test the backend locally.

4. Test the backend locally

After making changes to the backend, you can test it locally while it communicates with the remote frontend (on Vercel). This setup allows you to see real-time changes without deploying the backend to production.

- **Testing through the Vercel URL:**

Once your backend is running locally, open the frontend hosted on Vercel at <https://ai-marketer-v2-frontend.vercel.app> and test the application to ensure that the frontend is correctly communicating with the backend.

- **Check Logs:**

To verify that the backend is properly handling requests from the frontend, you can check the Docker logs. This will allow you to see incoming requests and ensure that the communication between the backend and frontend is working as expected.