

Guia de Instalação - Simplifika Post Backend em PHP

Sistema de Agendamento de Posts para Redes Sociais

Pré-requisitos

Seu servidor compartilhado deve ter:

- **PHP 7.4+** (geralmente já está instalado)
 - **MySQL 5.7+** ou **MariaDB 10.2+**
 - **Extensões PHP:** curl, json, pdo_mysql
 - **Acesso SSH** (recomendado)
 - **Cronjob** disponível
 - **Acesso FTP/SFTP** para upload de arquivos
-



Passo 1: Preparar o Banco de Dados

1.1 Acessar o MySQL via SSH

Conecte-se ao seu servidor via SSH:

```
ssh usuario@seu-servidor.com
```

Acesse o MySQL:

```
mysql -u seu_usuario -p
```

Digite sua senha quando solicitado.

1.2 Criar o Banco de Dados

Execute o script SQL fornecido (`schema.sql`):

```
mysql -u seu_usuario -p < schema.sql
```

Ou, se preferir manualmente:

```
CREATE DATABASE IF NOT EXISTS simplifika_post CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;  
USE simplifika_post;
```

1.3 Verificar a Criação

Verifique se o banco foi criado corretamente:

```
SHOW DATABASES;  
USE simplifika_post;  
SHOW TABLES;
```

Você deverá ver 7 tabelas: - users - user_profiles - connected_accounts - posts - hashtag_groups - alert_contacts - publication_logs

Passo 2: Preparar os Arquivos do Backend

2.1 Criar Diretório

Crie um diretório para o backend no seu servidor:

```
mkdir -p ~/public_html/api  
cd ~/public_html/api
```

2.2 Fazer Upload dos Arquivos

Faça upload dos seguintes arquivos PHP via FTP/SFTP:

```
config.php
Database.php
Auth.php
OAuthMeta.php
OAuthYouTube.php
index.php
cron.php
.env
```

Usando FTP: - Use FileZilla, WinSCP ou outro cliente FTP - Faça upload para
~/public_html/api/

Usando Git (se disponível):

```
git clone seu-repositorio-url .
```

2.3 Verificar Permissões

Certifique-se de que os arquivos têm as permissões corretas:

```
chmod 644 *.php
chmod 644 .env
chmod 755 .
```

Passo 3: Configurar Variáveis de Ambiente

3.1 Criar o Arquivo .env

Se não tiver feito upload do `.env`, crie-o manualmente:

```
nano .env
```

3.2 Preencher as Variáveis Essenciais

Banco de Dados:

```
DB_HOST=localhost
DB_USER=seu_usuario
DB_PASS=sua_senha
DB_NAME=simplifika_post
DB_PORT=3306
```

Segurança:

```
JWT_SECRET=gere_uma_chave_aleatoria_segura_aqui_minimo_32_caracteres  
SESSION_SECRET=outra_chave_aleatoria_segura_aqui
```

Aplicação:

```
APP_NAME=Simplifika Post  
APP_URL=https://seu-dominio.com  
APP_ENV=production  
APP_DEBUG=false
```

APIs Externas (obtenha as credenciais nos portais de desenvolvedor):

```
META_APP_ID=seu_app_id_aqui  
META_APP_SECRET=seu_app_secret_aqui  
YOUTUBE_CLIENT_ID=seu_client_id_aqui  
YOUTUBE_CLIENT_SECRET=seu_client_secret_aqui  
YOUTUBE_API_KEY=sua_chave_api_youtube_aqui
```

3.3 Salvar o Arquivo

Pressione `Ctrl+O` para salvar, depois `Ctrl+X` para sair do nano.



Passo 4: Testar a Instalação

4.1 Testar Conexão com Banco de Dados

Acesse via navegador:

```
https://seu-dominio.com/api/auth/me
```

Se receber um erro de autenticação (esperado), significa que a API está funcionando.

4.2 Testar Registro de Usuário

```
curl -X POST https://seu-dominio.com/api/auth/register \
-H "Content-Type: application/json" \
-d '{
  "email": "teste@exemplo.com",
  "password": "senha123",
  "name": "Teste"
}'
```

Você deverá receber um token JWT.

4.3 Testar Login

```
curl -X POST https://seu-dominio.com/api/auth/login \
-H "Content-Type: application/json" \
-d '{
  "email": "teste@exemplo.com",
  "password": "senha123"
}'
```



Passo 5: Configurar Cronjob para Agendamento

O Cronjob é essencial para que os posts sejam publicados automaticamente no horário agendado.

5.1 Via cPanel

1. Acesse **cPanel > Cron Jobs**
2. Clique em **Add New Cron Job**
3. Preencha os campos:
4. **Common Settings:** Every Minute
5. **Command:** `/usr/bin/php /home/seu_usuario/public_html/api/cron.php`
6. Clique em **Add New Cron Job**

5.2 Via SSH

```
crontab -e
```

Adicione a linha:

```
* * * * * /usr/bin/php /home/seu_usuario/public_html/api/cron.php
```

5.3 Verificar Cronjob

```
crontab -l
```

Você deverá ver a linha que acabou de adicionar.



Passo 6: Configurar HTTPS/SSL

HTTPS é essencial para segurança em produção.

6.1 Via cPanel (Recomendado)

1. Acesse **cPanel** > **SSL/TLS**
2. Clique em **Manage AutoSSL**
3. Selecione seu domínio
4. Clique em **Check & Install**

6.2 Manualmente (Let's Encrypt)

```
sudo certbot certonly --webroot -w ~/public_html -d seu-dominio.com
```



Passo 7: Integrar com Frontend

Seu frontend React precisa apontar para a API:

7.1 Configurar URL da API

No seu frontend, configure a URL base da API:

```
const API_URL = 'https://seu-dominio.com/api';
```

7.2 Fazer Requisições

Exemplo de login:

```
const response = await fetch(`${API_URL}/auth/login`, {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    email: 'usuario@exemplo.com',
    password: 'senha123'
  })
});

const data = await response.json();
const token = data.token;

// Armazene o token no localStorage
localStorage.setItem('token', token);
```

7.3 Usar Token em Requisições Autenticadas

```
const response = await fetch(`${API_URL}/posts`, {
  method: 'GET',
  headers: {
    'Authorization': `Bearer ${localStorage.getItem('token')}`
  }
});
```



Troubleshooting

Erro: "Cannot connect to database"

Solução: 1. Verifique as credenciais do MySQL no arquivo `.env` 2. Teste a conexão:

```
mysql -u seu_usuario -p -h localhost
```

 3. Verifique se o MySQL está rodando

Erro: "Call to undefined function curl_init()"

Solução: 1. A extensão curl não está instalada 2. Contate seu provedor de hospedagem para ativar 3. Ou use uma alternativa como `file_get_contents` com stream context

Erro: "Access denied for user"

Solução: 1. Verifique se o usuário e senha estão corretos 2. Verifique se o banco de dados foi criado 3. Verifique as permissões do usuário MySQL

Posts não estão sendo publicados

Solução: 1. Verifique se o Cronjob está configurado 2. Teste manualmente: `/usr/bin/php /caminho/para/cron.php` 3. Verifique os logs: `tail -f /var/log/cron` 4. Verifique se os tokens de acesso estão válidos

Erro de OAuth

Solução: 1. Verifique se o App ID e Secret estão corretos 2. Verifique se a URL de callback está configurada no painel de desenvolvedor 3. Verifique se o servidor está em HTTPS



Monitorar a Aplicação

Ver Logs de Erro

```
tail -f /var/log/php-errors.log
```

Ver Logs do Cronjob

```
tail -f /var/log/cron
```

Testar Cronjob Manualmente

```
/usr/bin/php /home/seu_usuario/public_html/api/cron.php
```

Backup do Banco de Dados

Fazer Backup Manual

```
mysqldump -u seu_usuario -p simplifika_post > backup_$(date +%Y%m%d_%H%M%S).sql
```

Restaurar Backup

```
mysql -u seu_usuario -p simplifika_post < backup_20240101_120000.sql
```

Agendar Backup Automático (Cron)

```
crontab -e
```

Adicione a linha:

```
0 2 * * * mysqldump -u seu_usuario -p'senha' simplifika_post > ~/backups/backup_$(date +%Y%m%d).sql
```

Checklist de Instalação

- ☐ Banco de dados criado
- ☐ Tabelas criadas com sucesso
- ☐ Arquivos PHP enviados
- ☐ Arquivo .env configurado
- ☐ Permissões de arquivo corretas
- ☐ Conexão com banco testada
- ☐ Registro de usuário funcionando
- ☐ Login funcionando
- ☐ Cronjob configurado
- ☐ HTTPS/SSL ativo
- ☐ Frontend integrado

- [] Testes de publicação bem-sucedidos
-

Próximos Passos

1. **Obter Credenciais das APIs:**
 2. Meta: <https://developers.facebook.com>
 3. YouTube: <https://console.cloud.google.com>
 4. TikTok: <https://developers.tiktok.com>
 5. **Preencher as Credenciais no .env**
 6. **Testar OAuth com cada plataforma**
 7. **Testar publicação de posts**
 8. **Submeter para App Review (Meta)**
-

Versão: 1.0

Data: 2024

Desenvolvido por: Manus AI