

# راهنمای پیکربندی AutoProjectManagement

---

دستورالعمل‌های کامل راه‌اندازی و پیکربندی



یکپارچه‌سازی



ذخیره‌سازی



امنیت



پیکربندی

# نمای کلی پیکربندی

راهنمای پیکربندی AutoProjectManagement دستورالعمل‌های جامعی را برای راه‌اندازی و سفارشی‌سازی تمام جنبه‌های محیط مدیریت پروژه شما ارائه می‌دهد.

## اهداف پیکربندی صحیح

- ✓ تضمین امنیت سیستم و حفاظت از داده‌ها
- ✓ بهینه‌سازی عملکرد برای محیط خاص شما
- ✓ فعال‌سازی یکپارچه‌سازی با سرویس‌های خارجی
- ✓ فراهم کردن ثبت وقایع قابل اعتماد برای عیب‌یابی
- ✓ پشتیبانی از محیط‌های استقرار مختلف

پیکربندی امنیتی



یکپارچه‌سازی GitHub



ذخیره‌سازی JSON



پیکربندی ثبت وقایع



تنظیمات محیطی



پیکربندی پروژه



مدیریت



اعتبارسنجی



# یكپارچه‌سازی GitHub

## </> مثال پیکربندی

```
{ "github": { "token": "ghp_XXXXXXXXXXXXXXXXXXXX",  
  "username": "yourusername", "default_repo":  
  "username/repository", "default_branch": "main",  
  "auto_sync": true, "sync_interval": 300 } }
```

## >\_ متغیرهای محیطی

ghp\_XXXXXXXXXXXXXXXXXXXX GITHUB\_TOKEN

yourusername GITHUB\_USERNAME

username/repository GITHUB\_DEFAULT\_REPO

## تنظیمات کلیدی ⚙️

فعال‌سازی همگام‌سازی خودکار با GitHub auto\_sync

زمان به ثانیه بین عملیات‌های همگام‌سازی sync\_interval

شاخه پیش‌فرض برای عملیات مخازن default\_branch

## 🔔 بهترین شیوه‌ها

✓ استفاده از توکن‌های دقیق با حداقل مجوزهای لازم

✓ ذخیره ایمن توکن‌ها، هرگز در کنترل نسخه ثبت نکنید

✓ تنظیم فواصل همگام‌سازی معقول برای جلوگیری از محدودیت‌های API

# پیکربندی امنیتی

## مؤلفه‌های امنیتی

### کلیدهای API

شناسه‌های منحصر به فرد برای کنترل دسترسی API

### توکن‌های JWT

توکن‌های وب JSON برای احراز هویت امن

### سیاست‌های CORS

کنترل‌های اشتراک‌گذاری منابع بین دامنه‌ها

### SSL/TLS

کانال‌های ارتباطی رمزنگاری شده

## </> مثال‌های پیکربندی

```
{ "security": { "jwt_secret_key": "your-super-secret-key-change-this", "jwt_algorithm": "HS256", "jwt_expiration_minutes": 60, "api_key_required": true, "api_key_header": "X-API-Key", "cors_allowed_origins": ["*"] } }
```

```
{ "security": { "ssl_cert_path": "/certs/server.crt", "ssl_key_path": "/certs/server.key" } }
```

## معماری امنیتی

### مجوزدهی

کنترل دسترسی‌ها

### احراز هویت

تأیید هویت کاربر

### محدودیت نرخ

جلوگیری از سوءاستفاده

### رمزنگاری

محافظت از داده‌ها

## گزینه‌های پیکربندی

تنظیم	نوع	توضیح
jwt_secret_key	رشته	کلید مخفی امضای JWT
jwt_algorithm	رشته	الگوریتم (HS256/RS256) JWT
api_key_required	بولین	نیاز به کلید API
cors_allowed_origins	لیست	منابع مجاز CORS

## بهترین شیوه‌ها

### اسرار قوی JWT

استفاده از اسرار طولانی و تصادفی برای امضای JWT

### انقضای توکن

تنظیم زمان انقضای معقول برای توکن‌های دسترسی

# پیکربندی ثبت وقایع

## معماری ثبت وقایع

### برنامه



منبع تمام رویدادهای ثبت وقایع

### پیکربندی ثبت وقایع



کنترل رفتار و قالب ثبت وقایع

### گرداننده‌های ثبت وقایع



کنسول، فایل، فایل چرخشی

### قالب ثبت وقایع



قالب استاندارد یا JSON

## گزینه‌های پیکربندی

تنظیم	نوع	توضیح
level	رشته	سطح ثبت وقایع (DEBUG, INFO, WARNING, ERROR)
format	رشته	رشته قالب ثبت وقایع
file_path	رشته	مسیر فایل ثبت وقایع
max_file_size	رشته	حداکثر اندازه فایل ثبت وقایع
json_format	بولین	استفاده از قالب JSON

### بهترین شیوه‌ها

چرخش منظم فایل‌ها

استفاده از سطوح ثبت وقایع مناسب

شامل مهر زمانی

استفاده از JSON در محیط تولید

## مثال‌های پیکربندی

### ثبت وقایع پایه کنسول



پیکربندی ساده برای محیط توسعه

```
{ "logging": { "level": "INFO", "format": "%(asctime)s - %(name)s - %(levelname)s - %(message)s" } }
```

### ثبت وقایع فایل با چرخش



ثبت وقایع فایل با قابلیت چرخش برای محیط تولید

```
{ "logging": { "level": "DEBUG", "file_path": "logs/autopproject.log", "max_file_size": "10MB", "backup_count": 5 } }
```

### ثبت وقایع ساختاریافته JSON



ثبت وقایع قابل خواندن توسط ماشین برای سیستم‌های نظارتی

```
{ "logging": { "level": "INFO", "json_format": true, "include_extra": true } }
```

# 📄 پیکربندی ذخیره‌سازی JSON

## 🏗️ معماری ذخیره‌سازی

### برنامه



تعامل با لایه ذخیره‌سازی JSON

### ذخیره‌سازی JSON



مدیریت تمام عملیات‌های داده JSON

### دایرکتوری JSonDataBase



شامل ورودی‌ها، خروجی‌ها، پشتیبان‌ها

## 🔧 گزینه‌های پیکربندی

تنظیم	نوع	توضیح
type	رشته	نوع ذخیره‌سازی (همیشه 'json')
json_path	رشته	فایل پیکربندی JSON اصلی
data_directory	رشته	دایرکتوری فایل‌های داده JSON
backup_enabled	بولین	فعال‌سازی پشتیبان‌گیری خودکار
max_file_size	عدد صحیح	حداکثر اندازه فایل JSON

### ★ بهترین شیوه‌ها

تنظیم محدودیت‌های اندازه فایل معقول ✓

فعال‌سازی پشتیبان‌گیری برای ایمنی داده‌ها ✓

استفاده از رمزگذاری UTF-8 ✓

سازماندهی منطقی فایل‌ها ✓

## </> مثال‌های پیکربندی

### 📁 ذخیره‌سازی JSON پایه

پیکربندی ساده با مسیرهای پیش‌فرض

```
{ "database": { "type": "json", "json_path": "autoproject.json", "data_directory": "JSonDataBase", "backup_enabled": true } }
```

### ➕ ساختار دایرکتوری سفارشی

مسیرهای سفارشی برای ذخیره‌سازی سازماندهی شده

```
{ "database": { "type": "json", "json_path": "config/project_data.json", "data_directory": "project_data", "inputs_path": "project_data/inputs", "outputs_path": "project_data/outputs", "backup_count": 10 } }
```

### 📄 ذخیره‌سازی JSON تولیدی

بهینه‌سازی برای محیط تولید

```
{ "database": { "type": "json", "json_path": "/opt/autoproject/data/main.json", "data_directory": "/opt/autoproject/data", "backup_enabled": true, "backup_count": 20, "max_file_size": 52428800 } }
```

# پیکربندی پروژه

## </> مثال پیکربندی

```
{ "project": { "base_path": "/path/to/project",  
  "data_path": "JJsonDataBase", "output_path":  
    "JJsonDataBase/OutPuts", "backup_path":  
    "project_management/PM_Backups", "max_file_size":  
    10485760, "allowed_extensions": [ ".py", ".js",  
      ".json", ".md" ], "max_workers": 8,  
  "timeout_seconds": 300, "enable_auto_commit": true,  
    "enable_risk_analysis": true,  
    "enable_progress_tracking": true,  
    "enable_github_integration": true } }
```

## 🏗️ ساختار پیکربندی

### ProjectConfig

تنظیمات و پارامترهای اصلی پیکربندی

### ProjectManager

راه اندازی دایرکتوری ها و اعتبارسنجی

### FileManager

مدیریت عملیات ها و پشتیبان گیری فایل ها

## ⚙️ تنظیمات کلیدی

### data\_path

مسیر دایرکتوری داده ها

### base\_path

دایرکتوری پایه پروژه

### allowed\_extensions

انواع فایل های مجاز

### max\_file\_size

حداکثر اندازه فایل (10MB)

### timeout\_seconds

زمان سنج عملیات

### max\_workers

حداکثر نخ های کارگر

### enable\_risk\_analysis

ویژگی های تحلیل ریسک

### enable\_auto\_commit

ثبت خودکار تغییرات

## 🔧 ویژگی های قابل تنظیم

### تحلیل ریسک

ارزیابی ریسک های احتمالی پروژه

### ثبت خودکار

ذخیره خودکار تغییرات در مخزن

### یکپارچه سازی GitHub

اتصال به مخازن GitHub

### پیگیری پیشرفت

نظارت بر پیشرفت پروژه

## ★ بهترین شیوه ها

محدود کردن پسوندهای فایل مجاز ✓

تنظیم محدودیت های اندازه فایل معقول ✓

تنظیم کارگران بر اساس منابع سیستم ✓

فعال سازی ثبت خودکار برای ایمنی ✓



# پیکربندی‌های محیطی خاص

## </> مثال‌های محیطی

### توسعه



اشکال‌زدایی: فعال

ثبت وقایع: DEBUG

اشکال‌زدایی: فعال

```
{ "api": { "host": "127.0.0.1", "port": 8000, "debug": true, "reload": true }, "logging": { "level": "DEBUG" } }
```

### تولید



محدودیت نرخ: سختگیرانه

ثبت وقایع: INFO

SSL: فعال

```
{ "api": { "host": "0.0.0.0", "port": 8080, "debug": false }, "security": { "ssl_cert_path": "/certs/server.crt" }, "logging": { "level": "INFO" } }
```

### تست



اشکال‌زدایی: فعال

ذخیره‌سازی: در حافظه

اشکال‌زدایی: فعال

```
{ "api": { "host": "127.0.0.1", "port": 8001, "debug": true }, "database": { "json_path": ":memory:" }, "project": { "enable_github_integration": false } }
```

## ماتریس پیکربندی

ویژگی	توسعه	تولید	تست
اشکال‌زدایی API	فعال	غیرفعال	فعال
نوع ذخیره‌سازی	فایل‌های JSON	فایل‌های JSON	فایل‌های JSON
سطح ثبت وقایع	DEBUG	INFO	DEBUG
یکپارچه‌سازی GitHub	فعال	فعال	غیرفعال
SSL مورد نیاز	خیر	بله	خیر
محدودیت نرخ	منعطف	سختگیرانه	غیرفعال

## تشخیص محیط



### بارگذاری تنظیمات

اعمال پیکربندی مناسب



### تطابق مقدار

شناسایی نوع محیط



### بررسی

### ENVIRONMENT

خواندن متغیر محیطی

## تفاوت‌های کلیدی

SSL مورد نیاز در تولید

حالت اشکال‌زدایی در توسعه/تست

محدودیت نرخ سختگیرانه در تولید

ثبت وقایع مفصل در توسعه/تست



# اعتبارسنجی و مدیریت پیکربندی

## فرآیند اعتبارسنجی

۱ بارگذاری پیکربندی  
خواندن و تجزیه فایل‌های پیکربندی

۲ اعتبارسنجی فیلدهای مورد نیاز  
بررسی وجود تنظیمات اجباری

۳ اعتبارسنجی انواع داده  
تأیید انواع داده صحیح برای تمام مقادیر

۴ اعتبارسنجی تنظیمات امنیتی  
بررسی پارامترهای امنیتی برای ریسک‌ها

۵ اعتبارسنجی وابستگی‌ها  
تأیید اتصالات سرویس‌های خارجی

## ابزارهای مدیریت

ابزارهای  
CLI

مدیریت  
برنامه‌نویسی  
پیکربندی

API  
پایتون

مدیریت  
برنامه‌نویسی  
پیکربندی

سیستم  
پشتیبان

پشتیبان‌گیری  
خودکار پیکربندی

بارگذاری  
مجدد

به‌روزرسانی‌های  
پیکربندی زنده

## قوانین اعتبارسنجی

فیلدهای مورد  
نیاز موجود

نحو JSON  
معتبر

مسیرهای فایل  
معتبر

انواع داده  
صحیح

درگاه‌های موجود

تنظیمات امنیتی

## فرآیند بارگذاری مجدد زنده



اعمال

به‌روزرسانی برنامه در حال اجرا



اعتبارسنجی

فرآیند اعتبارسنجی پیکربندی



تشخیص تغییر

اعلان تغییر فایل سیستم



به‌روزرسانی پیکربندی

مدیر فایل پیکربندی را ویرایش  
می‌کند

## دستورات مدیریت

```
# نمایش پیکربندی # print_config() # اعتبارسنجی پیکربندی فعلی # config.validate() # پایتون API #
CLI --config بارگذاری مجدد پیکربندی از فایل‌ها # آرگومان‌های # reload_config() # فعلی
# --generate # اعتبارسنجی پیکربندی # --validate # مشخص کردن فایل پیکربندی # config.json
# مشخص کردن محیط # --env development - تولید نمونه پیکربندی
```

## پشتیبان‌گیری و بازیابی

### دستورات بازیابی

```
cp config.backup.json config.json
```

```
python -m
autoprojectmanagement.configuration
validate
```

```
python -m
autoprojectmanagement.configuration
restore
```

### دستورات پشتیبان‌گیری

```
cp config.json config.backup.json
```

```
cp config.json config.backup.%(date
+Y%m%d_%H%M%S).json
```

```
python -m
autoprojectmanagement.configuration
backup
```

# عیب‌یابی و مرجع سریع

## ⚠ مسائل و راه‌حل‌های رایج

### ❌ فایل مورد نیاز از دست رفته

افزودن پارامتر پیکربندی گمشده

### 🔗 JSON نامعتبر

استفاده از اعتبارسنج JSON برای بررسی نحو

### 🔍 فایل JSON یافت نشد

بررسی مسیرها و مجوزهای فایل

### 🔌 پورت نامعتبر

تغییر به پورت موجود (1-65535)

### 🛡 خطای گواهی SSL

بررسی مسیرها و اعتبار گواهی‌ها

### 🔑 توکن GitHub نامعتبر

تولید مجدد توکن GitHub با مجوزهای مناسب

## ☰ چک‌لیست پیکربندی

🔗 نحو JSON معتبر است

✅ تمام فایل‌های مورد نیاز موجود هستند

📁 مسیرهای فایل JSON صحیح هستند

🐙 توکن GitHub مجوزهای صحیح دارد

🛡 تنظیمات امنیتی به درستی پیکربندی شده‌اند

## ❓ دریافت کمک

### 📄 مستندات

📄 راهنمای نصب

🔗 API مرجع

📄 راهنمای پیکربندی

### 👥 پشتیبانی جامعه

📄 صفحات ویکی

💬 بحث‌ها

🐛 گزارش مشکلات GitHub

### 🎧 پشتیبانی حرفه‌ای

✉ support@autoprojectmanagement.com

📄 docs@autoprojectmanagement.com

🛒 consulting@autoprojectmanagement.com

## ➤ دستورات تشخیصی

### 🔧 تشخیص‌های سریع

```
python -m json.tool config.json
```

```
python -c "from autoproject_configuration import config; config.validate()"
```

```
python -c "from autoproject_configuration import config; print(f'Storage: {config.database.type}')" 
```

```
python -c "from autoprojectmanagement.services.github_integration import GitHubIntegration; print('GitHub ready')"
```