Configuration

- Home
- Components
  - InfoSymbolServer.Domain
  - InfoSymbolServer.Infrastructure
    - Data Access
    - Background Jobs
    - Notifications
  - InfoSymbolServer.Application
  - InfoSymbolServer.Presentation
  - InfoSymbolServer
- Versioning
- **Configuration** *(current)*
- Deployment

Overview

InfoSymbolServer uses the standard .NET configuration system with `appsettings.json` files. This page explains the configuration options available in the application and how to customize them for different environments.

Configuration File Structure

The main configuration file is `appsettings.json`. Here's an overview of its structure:

```
{
  "Serilog": {
    // Logging configuration
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    // Database connection strings
  },
  "BackgroundJobs": {
    // Quartz.NET job schedules
  },
  "Notifications": {
    // Notification service settings
  }
}
```

Logging Configuration (Serilog)

InfoSymbolServer uses Serilog for structured logging. The configuration defines log levels, outputs, and formatting:

```
"Serilog": {
  "Using": [ "Serilog.Sinks.Console" ],
  "MinimumLevel": {
    "Default": "Information"
  },
  "WriteTo": [
    {
      "Name": "Console",
      "Args": {
        "outputTemplate": "[{Timestamp:HH:mm:ss} {Level:u3}] {Message:lj}{NewLine}{Exception}"
      }
    }
  ],
  "Enrich": [ "FromLogContext", "WithMachineName", "WithThreadId" ]
}
```

Key Settings:

- **Using**: List of Serilog sinks to use (e.g., Console, File)
- **MinimumLevel**: Defines the minimum log level to capture
- **WriteTo**: Configuration for output destinations
- **Enrich**: Additional context to include in log entries

Database Connection

Database connection strings are defined in the `ConnectionStrings` section:

```
"ConnectionStrings": {
  "DefaultConnection": "Host=db;Database=infosymboldb;Username=postgres;Password=postgres;IncludeErrorDetail=true"
}
```

Key Settings:

- **Host**: Database server hostname (e.g., `db` for Docker, `localhost` for local development)
- **Database**: Name of the database
- **Username/Password**: Database credentials

- **IncludeErrorDetail**: Whether to include detailed error information (useful for development)

Environment-Specific Configurations:

For production environments, you should use environment variables or secrets management:

```
# Set connection string via environment variable
export ConnectionStrings__DefaultConnection="Host=production-
db;Database=infosymboldb;Username=app_user;Password=secure_password;"
```

Background Jobs

The `BackgroundJobs` section configures Quartz.NET job schedules:

```
"BackgroundJobs": {
  "BinanceSymbolSyncJobSchedule": "0 0/3 * * * ?"
}
```

Key Settings:

- **BinanceSymbolSyncJobSchedule**: Cron expression for the Binance symbol synchronization jobs

Cron Expression Format

The cron expression follows the format:

For example:

- `0 0/3 * * * ?`: Run at 0 seconds, every 3 minutes
- `0 0 */2 * * ?`: Run at the top of every even-numbered hour
- `0 0 0 * * ?`: Run at midnight every day

All jobs are scheduled using the same cron expression defined in the configuration.

Notifications

The `Notifications` section configures the notification services used by the application:

> **Note**: Notification configuration is optional. If you don't need to receive notifications about symbol changes or system issues, you can leave these settings empty or omit them entirely. The system will function normally without notifications.

```
"Notifications": {
  "Telegram": {
    "BotToken": "",
    "ChatIds": [ "" ],
    "IncludeDetailedInfo": true,
    "MaxSymbolsPerMessage": 5
  },
  "Email": {
    "SmtpServer": "",
    "SmtpPort": 587,
    "UseSsl": true,
    "SenderEmail": "",
    "SenderName": "",
    "Username": "",
    "Password": "",
    "Recipients": [
      ""
    ]
  }
}
```

Telegram Notifications

The `Telegram` section configures the Telegram bot used for sending regular notifications:

Key Settings:

- **BotToken**: The API token for your Telegram bot (obtained from BotFather)
- **ChatIds**: IDs of chats where notifications will be sent
- **IncludeDetailedInfo**: Whether to include detailed information in notifications
- **MaxSymbolsPerMessage**: Maximum number of symbols to include in a single message

Email Notifications

The `Email` section configures the email service used for emergency notifications:

Key Settings:

- **SmtpServer**: The SMTP server address
- **SmtpPort**: The SMTP server port (typically 587 for TLS)
- **UseSsl**: Whether to use SSL/TLS for the connection
- **SenderEmail**: The email address used as the sender
- **SenderName**: The display name for the sender
- **Username**: The username for SMTP authentication
- **Password**: The password for SMTP authentication

- **Recipients**: Array of email addresses that will receive notifications