

Hangfire.SqlServer

August 17, 2026

Hangfire.SqlServer

Kind: Service

Source: Pams/API/Pams.API/Pams.API.csproj (line 1)

Part of: Pams

NuGet package dependency

Hangfire.SqlServer is a NuGet package dependency in Pams.API that adds SQL Server-backed storage for Hangfire jobs. When configured by the API host, Hangfire stores queued work, job state, and worker coordination data in SQL Server.

Diagram

```
sequenceDiagram
    participant Client
    participant API as Pams.API
    participant Hangfire
    participant SQL as SQL Server
    participant Worker as Hangfire Worker

    Client->>API: Submit work request
    API->>Hangfire: Enqueue background job
    Hangfire->>SQL: Store job and state
    Worker->>SQL: Poll queued jobs
    Worker->>Worker: Execute job handler
    Worker->>SQL: Update job state
```

Usage

```
async function submitBackgroundWork(payload: Record<string, unknown>) {
    const response = await fetch("/api/jobs", {
        method: "POST",
        headers: {
            "Content-Type": "application/json",
```

```
    },  
    body: JSON.stringify(payload),  
  });  
  
  if (!response.ok) {  
    throw new Error("The API could not queue the background job.");  
  }  
  
  return response.json();  
}  
  
await submitBackgroundWork({  
  jobType: "process-record",  
  recordId: "record-id",  
});
```

AI Coding Instructions

- Treat `Hangfire.SqlServer` as server-side infrastructure; client code should request work through API endpoints rather than access Hangfire storage directly.
- Keep Hangfire SQL Server configuration in the API startup and dependency-injection path that configures background job storage.
- Use job handlers with serializable arguments; avoid passing request objects, database contexts, or open connections into queued jobs.
- Check SQL Server connectivity and migration permissions when adding or changing Hangfire storage configuration.